A K-MEANS GEOMETRIC SMOTE WITH DATA COMPLEXITY ANALYSIS FOR IMBALANCED DATASET

NUR ATHIRAH BINTI AZHAR

MASTER OF SCIENCE (INTELLIGENT SYSTEM)
UNIVERSITI UTARA MALAYSIA
2023

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

NUR ATHIRAH AZHAR

calon untuk Ijazah **MASTER OF SCIENCE (INTELLIGENT SYSTEM)**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"A K-MEANS GEOMETRIC SMOTE WITH DATA COMPLEXITY ANALYSIS FOR IMBALANCED DATASET"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***07 Mac 2023***.
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: ***07 March 2023.***

Pengerusi Viva: *(Chairman for VIVA)*	**Assoc. Prof. Ts. Dr. Nor Laily Hashim**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Dr. Samry @ Mohd Shamrie Sainin**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Dr. Azizi Ab Aziz**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Dr. Muhammad Syafiq Mohd Pozi**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Amiza Mohamed Din**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Adam Jatowt**	Tandatangan *(Signature)*

Tarikh:
(Date) ***07 March 2023***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Banyak set data kelas binari dalam aplikasi kehidupan sebenar dipengaruhi oleh masalah ketidakseimbangan kelas. Kerumitan data seperti contoh hingar, pertindihan kelas dan masalah perpecahan kecil diperhatikan memainkan peranan penting dalam menghasilkan prestasi pengelasan yang lemah. Kerumitan ini wujud seiring dengan masalah ketidakseimbangan kelas. Teknik Terlebih Sampel Minoriti Sintetik (SMOTE) ialah kaedah yang terkenal untuk mengimbangi semula bilangan contoh dalam set data tidak seimbang. Walau bagaimanapun, teknik ini tidak dapat menangani kerumitan data dengan berkesan dan mempunyai keupayaan untuk meningkatkan tahap kerumitan. Oleh itu, pelbagai varian SMOTE telah dicadangkan untuk mengatasi kelemahan SMOTE. Tambahan pula, tiada kajian sedia ada yang mengenal pasti kolerasi antara ukuran kerumitan N1 dan ukuran pengelasan seperti min geometri (G-Mean) dan Skor-F1. Kajian ini bertujuan: (i) untuk mengenal pasti ukuran kerumitan yang sesuai dan mempunyai kolerasi dengan ukuran prestasi, (ii) untuk mencadangkan varian SMOTE baharu yang menggabungkan ukuran kerumitan semasa tugas penjanaan data sintetik, dan (iii) untuk menilai varian SMOTE yang dicadangkan dalam jangka prestasi klasifikasi. Siri eksperimen telah dijalankan untuk menilai prestasi klasifikasi yang berkaitan dengan G-Mean dan F1-Score serta pengukuran kerumitan N1 bagi varian SMOTE penanda aras dan KM-GSMOTE. Prestasi KM-GSMOTE dinilai pada 6 set data binari penanda aras daripada repositori UCI. KM-GSMOTE merekodkan peratusan tertinggi bagi purata perbezaan G-Mean (22.76%) dan F1-Score (15.13%) untuk pengelas SVM. Korelasi antara langkah pengelasan dan ukuran kerumitan N1 telah diperhatikan daripada keputusan eksperimen. Sumbangan kajian ini adalah (i) pengenalan KM-GSMOTE yang menggabungkan pengukuran kerumitan dengan pemilihan model untuk memilih model dengan prestasi pengelasan terbaik dan nilai kerumitan yang lebih rendah dan (ii) pemerhatian hubungan antara prestasi pengelasan dan ukuran kerumitan di mana sekiranya ukuran kerumitan N1 berkurangan, kebarangkalian untuk memperoleh prestasi pengelasan yang memberangsangkan meningkat.

Kata Kunci: SMOTE, Ketidakseimbangan kelas, Kerumitan data, Pensampelan berlebihan, Klasifikasi.

Abstract

Many binary class datasets in real-life applications are affected by class imbalance problem. Data complexities like noise examples, class overlap and small disjuncts problems are observed to play a key role in producing poor classification performance. These complexities tend to exist in tandem with class imbalance problem. Synthetic Minority Oversampling Technique (SMOTE) is a well-known method to re-balance the number of examples in imbalanced datasets. However, this technique cannot effectively tackle data complexities and has the capability of magnifying the degree of complexities. Therefore, various SMOTE variants have been proposed to overcome the downsides of SMOTE. Furthermore, no existing study has yet to identify the correlation between N1 complexity measure and classification measures such as geometric mean (G-Mean) and F1-Score. This study aims : (i) to identify the suitable complexity measures that have correlation with performance measures, (ii) to propose a new SMOTE variant which is K-Means Geometric SMOTE (KM-GSMOTE) that incorporates complexity measures during synthetic data generation task, and (iii) to evaluate KM-GSMOTE in term of classification performance. Series of experiments have been conducted to evaluate the classification performances related to G-Mean and F1-Score as well as the measurement of N1 complexity of benchmark SMOTE variants and KM-GSMOTE. The performance of KM-GSMOTE was evaluated on 6 benchmark binary datasets from the UCI repository. KM-GSMOTE records the highest percentage of average differences of G-Mean (22.76%) and F1-Score (15.13%) for SVM classifier. A correlation between classification measures and N1 complexity measures has been observed from the experimental results. The contributions of this study are (i) introduction of KM-GSMOTE that combines complexity measurement with model selection to pick models with the best classification performance and lower complexity value and (ii) observation of connection between classification performance and complexity measure, showing that as N1 complexity measure decreases, the likelihood of obtaining a substantial classification performance increases.

Keywords: SMOTE, Class imbalance, Data complexity, Oversampling, Classification.

Acknowledgements

I would like to express my appreciation and gratitude to everyone who has contributed in completing this thesis. It was my pleasure to study under Dr. Muhammad Syafiq Mohd Pozi's supervision. It is not enough to thank you very much to him for his guidance to help me to achieve my goal. Without his valuable support, my thesis would not have been possible. I would like to express my thanks to my co-supervisor Ms. Aniza Mohamed Din and Prof Adam Jatowt for their comments which help to improve my work. I would like also to thank my parents, Azhar Kamis and Salmiah Morad as well as all of my siblings, Nurul Anis Azhar, Muhammad Arif Azimi and Nur Afifah Azhar for their love and support. My goal would not have been achieved without them. I dedicate this work to my parents. I am very grateful to Dr Mohamad Farhan Mohamad Mohsin and Dr Azizi Ab Aziz. They were very kind during the proposal defense and during the period of the correction. Additionally their comments have helped to improve this work. I had a very enjoyable study at Universiti Utara Malaysia (UUM). Not only, does it has a beautiful natural environment but the university also has helpful staff. Finally, I would like to thank all of my friends, especially Nor Yusra Md Rosele and Nurul Musfirah Murad as well as my fiancé Muhammad Muaz Mohd Nahar for their encouragement during my study.

Table of Contents

List of Tables

List of Figures

CHAPTER ONE
INTRODUCTION

The definition of an imbalanced classification problem can be described whenever the number of representatives in one class is larger than the other classes, thus the distribution of the dataset is skewed. Due to insufficient representation in minority class, the underlying pattern of the minority class is hardly to be captured and properly generalized (Tanha et al., 2020; Patel et al., 2020; Yan et al., 2019). As a result, the classifier tends to show bias towards the majority class that has prevalent examples (Krawczyk, 2016; Haixiang et al., 2017). In real-life application like facial recognition used in healthcare, bias is insidious especially in terms of gender and racial bias such that the health disparities faced by under-represented minority class could be widen besides jeopardizing equality of treatment Larrazabal et al. (2020).

In order to properly understand and solve imbalanced classification problem, the data intrinsic characteristics must be identified as they might inflict difficulties for machine learning algorithm. In the literature, it is found that there are three types of data complexities that needs to be considered when performing learning task on imbalanced dataset (Garcia et al., 2015; Yuan et al., 2020; Barella et al., 2021). Those data complexities are noise examples, class overlap and small disjuncts. Noise examples appears when there are examples located in the opposite class label and deemed useless since no information about the class label is being provided. Class overlap problem occurs when two or more points from different classes in the data space, are found in an ambiguous region between these classes. On the other hand, the existence of subconcepts (regardless of the class) leads to a phenomena called small disjuncts where these disjuncts usually under-represent the concept of a class as well as covering only a small part of the dataset.

It is necessary to verify whether or not the data at hand have an imbalanced classification problem. This can be verified either at the data level itself, i.e., by measuring the class imbalanced ratio, because if the class imbalanced ratio is very high, it is very likely that the data is heavily unbalanced. Another method of verification is the classification performance itself. Performance metrics such as F-measure (Parambath et al., 2014) and G-Mean (Tang et al., 2008) can be used to indicate how balanced the classification performance is for any newly derived classification model.

Once the degree of imbalanced classification problem is properly defined, a mitigation plan can properly be drafted. If the definition is represented as the result of the classification performance itself (by F-Measure or G-Mean), then one can try to improve the classification performance through an exhaustive model selection task. Model selection can be defined as the selection of a good machine learning model either from different models types like Logistic Regression, Gaussian Naive Bayes and Support Vector Machine (SVM) or from the same model type considering features, hyperparameters, model parameters and others (Du et al., 2020; Gao et al., 2020a; Adedigba et al., 2021). In this way, the classification performance for the minority class can be increased, without sacrificing the overall classification performance (Pozi et al., 2016, 2022; Richhariya and Tanveer, 2020). On the other hand, if the definition is represented as the imbalance ratio of the dataset, then one can try to solve the issue by acquiring more data for the less represented class. Since the data is usually difficult to be collected or is impossible to obtain, the class distribution can be balanced by removing redundant examples (undersampling) from the existing dataset or by adding synthetic examples (oversampling) through various resampling techniques (Chawla et al., 2002; Bunkhumpornpat et al., 2009; Wang et al., 2014).

Another option aside from data-level approach for solving imbalanced datasets could

(safe area) or near the borderline of minority class. Besides, SMOTE also can be extended through integration of SMOTE with other pre-processing techniques such as filtering, clustering and weighting algorithms. Combination of clustering algorithm with weighting algorithm has been quite famous in the recent studies due to the fact that not only a proper weight can be assigned to every seed examples based on their characteristics but also the generated synthetic examples will be contained in the designated region (Barua et al., 2012; Nekooeimehr and Lai-Yuen, 2016; Tao et al., 2020; Wang and Wang, 2021).

This study proposes a new SMOTE variant known as KM-GSMOTE which utilizes *k*-means clustering algorithm and Geometric SMOTE, an improvement of the SMOTE data generation mechanism. This technique incorporates data complexity analysis into the observation of classification performance to bring a new perspective to the performance evaluation of SMOTE domain. Before doing so, it is important to determine the correlation between complexity and classification measures is important to find out how the former would affect the latter which can be used to predict the classification performance based on complexity analysis itself.

1.1 Problem Statement

To the best of our knowledge, there is no measurement that has been proposed to estimate the degree of noise in a dataset. Koziarski et al. (2019) changes the original data used in the experiment by setting the noise level and then artificially reproducing it, which allows the noise level to be determined. The same is true for identifying small disjuncts, but only recently Datta et al. (2017) introduced a technique for this purpose that does not rely on a rule-based learning scheme and is based on the assumption that subclusters within classes correlate with disjuncts. The fundamental element of this technique is that the Breadth First Search (BFS) determines the unique disjuncts

in each class. The degree of class overlap, on the other hand can be estimated using some data complexity measures that address this issue.

Initially, the assessment of data characteristics as well as the measurement of degree of complexity are not easy to acquire especially for imbalanced datasets due to the consideration of complex concept and the need to gather more information. Therefore, data complexity measures have been proposed by Ho and Basu (2002) to measure and approximate the complexity of a dataset besides giving an insight into its intrinsic characteristics. These works also were extended by Barella et al. (2021) to improve the effectiveness of the measurements so that the evaluation of the data complexity is much more better. Complexity measures are becoming more significant in the studies associated with imbalanced classification problem as imbalance ratio could not adequately portray the existing difficulties lay in imbalanced datasets (Lu et al., 2019). Imbalance ratio is typically regarded as the prime issue in imbalanced classification problem. Comparison analysis based on imbalance ratio is however rather simplistic because datasets in real-life are affected by other factors like the above-mentioned data complexities (Das et al., 2018; Garcia et al., 2015).

The basic and commonly used strategy in solving imbalanced datasets is to balance the class distribution in the first place, before the learning task taking place. However, most of the time, it is not easy to acquire more data due to the nature of data source. SMOTE (Chawla et al., 2002) is one of the most well-known oversampling techniques where interpolation between minority examples takes place, introducing synthetic examples to rebalance the class distribution in a dataset. The good quality found in the synthetic examples generated using SMOTE is that they provide additional useful information about minority class which could avoid over-fitting problem from appearing instead of merely giving duplicated information like random oversampling does.

Despite all that, SMOTE suffers from several shortcomings such as it might exacerbate the noisy environment as well as the class overlap, thus increasing the complexity of the dataset. Moreover, SMOTE is only capable of properly addressing between-class imbalance problem, leaving the within-class imbalance problem (small disjuntcs) unsolved (Fernández et al., 2018). All of these issues arise due the fact that SMOTE randomly chooses minority examples for interpolation without taking into consideration the location of these examples since they might happen to be in the region of majority class. This situation causes the algorithm to blindly generate synthetic examples regardless of the characteristics of distribution (Liang et al., 2020). Hence, various modifications (Douzas and Bacao, 2019; Chen et al., 2020) and extensions (Barua et al., 2012; Sáez et al., 2015) of SMOTE have been introduced to overcome the limitations of SMOTE besides alleviating any of the data complexities incorporated in the datasets.

Performance evaluation of SMOTE variants is usually based on performance measures such as accuracy, precision, recall and F1-score which depend on the number of correct or incorrect classifications of a given class (Douzas and Bacao, 2019; Adhikari et al., 2020; Wang and Wang, 2021). The variants focus on increasing the value of these metrics so that the metrics can show how well the implemented oversampling strategy improves the learning and classification task. Despite the variants of SMOTE are capable of handling some data complexity, none of the variants have ever considered complexity measures to provide insight into the impact of the oversampling strategy on the degree of complexity. Studies by Barella et al. (2018) suggests that there is a correlation between complexity measures and performance measures. This means the former can be used prior to the latter to approximate the classification performance which will be very useful for evaluating the performance of SMOTE variants.

In conclusion, this study uses one of the data-level approaches, oversampling technique by proposing a new SMOTE variant, KM-GSMOTE to tackle the class imbalance problem instead of algorithm-level approaches. The former requires lower computational cost compared to the latter and is also versatile without being limited to a particular classifier, as is the case with the latter (Ali et al., 2019a). KM-GSMOTE combines k-means clustering and a state-of-the-art SMOTE variant, Geometric SMOTE which allows the allocation of suitable number of synthetic minority examples to be generated in a cluster and also formation of informative synthetic minority examples by taking into consideration the nearest neighbours selection strategy. Moreover, this study aims to identify a suitable complexity measure to further investigate its relationship with performance measures. After establishing the correlation between these two measures, the identified complexity measures are used as the basis of classification performance for KM-GSMOTE. Lastly, the evaluation procedure for this technique is performed to analyze the classification performance.

1.2 Research Questions

From the preceding discussion, this study will clarify the following research questions:

1. Which complexity measures correlate with classification measures?
2. How to design KM-GSMOTE to solve imbalanced classification problem?
3. How to design a parameter selection model for the proposed SMOTE variant?
4. What is the performance of KM-GSMOTE?

1.3 Research Objectives

This study embarks on the following objectives:

1. To identify the suitable complexity measures that have correlation with performance measures.

2. To propose KM-GSMOTE that incorporates complexity measures during synthetic data generation task.
3. To evaluate KM-GSMOTE in term of classification performance.

1.4 Scope

Imbalanced classification problem poses a serious challenge to the conventional machine learning classifier. Therefore, this study will be using current benchmark imbalanced datasets to evaluate the performance of learning algorithm, in term of classification accuracy of both majority and minority class. Furthermore, imbalanced class datasets can be categorized into two types; binary and multi-class but the focus of this study is set on the former one. 5 binary imbalanced binary class datasets that have been identified, all of them with various imbalance ratio values; the lowest is 8.6 while the highest is 33.74. All datasets are collected from one of the famous repositories among machine learning researchers, UC Irvine Machine Learning Repository.

Then, the imbalanced classification problem will be tackled using data-level approach that utilizes resampling technique instead of algorithmic approach. Although there are two types of resampling techniques which are undersampling and oversampling, this study only focuses on oversampling technique or to be more specific, SMOTE algorithm. The choice is made because SMOTE is able to re-balance the class distribution by generating synthetic examples through interpolation between minority examples (Chen et al., 2021). For this reason, the number of minority examples is increased to the point this class is not more outnumbered by the majority class and the dataset is finally balanced. Moreover, SMOTE algorithm might enhance the generalization capability of classifier since more information provided through generation of synthetic examples (Fernández et al., 2018).

Instead of merely relying on the performance measures to analyze the performance of a technique, this study will incorporate complexity measures to get more understanding on how oversampling technique affects data complexity. These measures helps to estimate the the difficulties of imbalanced datasets (Barella et al., 2021) especially for this study when observing the difference in data complexity after SMOTE variants have been applied.

1.5 Significance

The significance of this study is twofold: which are to the body of knowledge (theoretical contribution) and application perspective. From a theoretical perspective, this study will contribute to the knowledge of the correlation between complexity measures and performance measures in the context of SMOTE. Previous researches have not linked the two measures in analyzing the performance of SMOTE variants as they consider the classification performance through performance measures. Moreover, quantifying data complexity by using complexity measures provides a broader perspective in the analysis of SMOTE. Apart from that, the proposed new SMOTE variant takes degree of data complexity into consideration such that it aims to reduce the complexity and improve the classification performance.

From an application perspective, in real-world applications such as healthcare (healthy patients vs. cancer patients), finance (valid transactions activity vs. unauthorized or malicious), or software quality (good vs. buggy), there is often a class imbalance problem where the importance of minority class outweighs that of majority class. To make matters worse, data complexities ususally hinders the learning and classification tasks even more than the class imbalance problem leading to a degradation in classification performance. Therefore, implementation of KM-GSMOTE with data complexity analysis would conceivably improve the classifier's ability to classify minority classes

in these datasets such as credit card fraud, cancer patients, or the occurrence of natural disasters. The classifier is able to construct a generalized classification model since this technique feeds effective synthetic minority examples that provides new information to the classifier. Ultimately, the detection or prediction task becomes more reliable and precise as the minority class significance is improved, particularly through the use of KM-GSMOTE.

In addition, KM-GSMOTE gives information about the intrinsic characteristics of the data through data complexity measures that indicate how difficult the classification task is. By demonstrating a relationship between data complexity and classification performance, the data scientist can further process the data to reduce complexity, which should lead to better classification performance. The data scientist can decide a good parameter selection model by tuning the number of clusters during the k-means clustering phase to identify the model that can significantly minimize complexity.

1.6 Summary

Complexity measures has been found to be crucial in portraying the intrinsic characteristics as well as the difficulties of a dataset. For this reason, the measures are being applied in this thesis in order to observe the impact of oversampling strategies, in this case SMOTE and its variants have on the complexity of the dataset. By doing so, the correlation between complexity measures and performance measures which typically used to evaluate the classification performance can be observed. The discussion on the correlation might help to provide knowledge about what improvement of oversampling strategy adopted in SMOTE can be made in order to propose a new variant that able to tackle the complexity of any imbalanced datasets.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This chapter elaborates more on the literature review of the study. Since the main discussion of this study is related to data complexity and SMOTE, this chapter will present the existing works proposed to tackle data complexities like noise, class overlap and small disjuncts which are commonly associated with class imbalanced problem. This chapter also discuss about the extensions and modifications of SMOTE proposed by previous research studies. The algorithms applied in the techniques will be identified to find the merits and the limitations thus disclosing the gap. The discussion in this chapter is arranged accordingly based from the identified problems and objectives stated in Chapter 1.

This study involves class imbalanced problem study; therefore, the discussion starts by presenting the data complexities in imbalanced datasets in Section 2.2. Section 2.3 explains about the techniques proposed to address imbalanced datasets. Lastly, this chapter ends with a summary that talk about the gap of the study in Section 2.4.

2.2 Data Complexities in Imbalanced Dataset

In general, the class imbalanced problem can be observed in the form of causal effect, that is, when a particular class distribution is skewed in one direction, and the classifier fails to appropriately capture the underlying pattern hidden in the minority class, leading to poor classification performance on minority class' examples. While there are many reasons of why the classifier performs poorly on any given dataset, from the perspective of data, it is found that there are three other sources besides class imbalance problem that are usually accompanying imbalanced dataset, thus contributing to

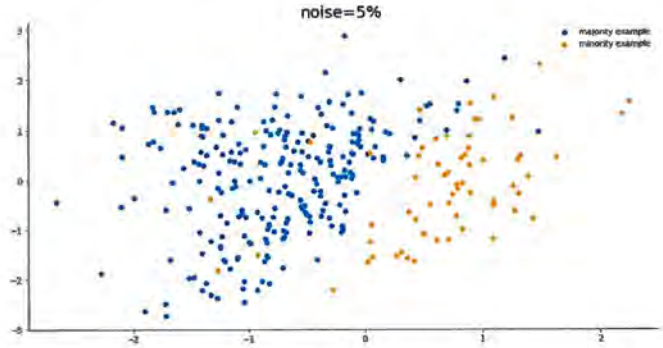
the poor classification performance. Those sources are noise examples, class overlap, and small disjuncts where they are known as data complexities in imbalanced dataset. As stated in (Kong et al., 2019), analyzing data complexity is crucial in determining a suitable approach to solve the problem. Each data complexity is properly discussed in this section.

2.2.1 Noise Examples

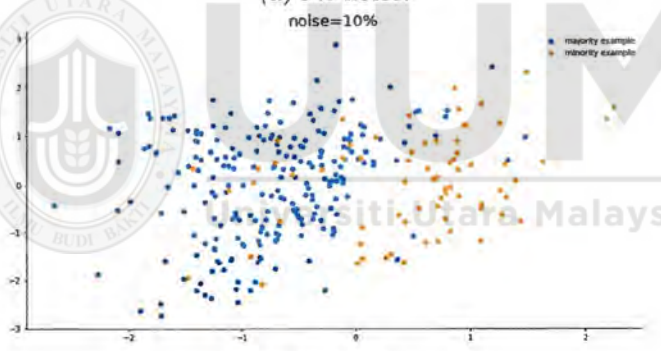
Noise examples have been a challenge to be addressed by researchers as they are common to be found in real-word datasets. These kind of examples can poorly affect the classifier performance in terms of prediction accuracy rates (Garcia et al., 2015). The existence of an example with a different class label within another class region is what considered as noise examples or label noise as presented in Figure 2.1 where it indicates that there could be various degree of noise in an imbalanced dataset. Label noise that linked to anomalies, abnormalities and novelties is another type of noisy data apart from feature noise (Koziarski et al., 2019) such that the latter are associated with missing, incorrect and incomplete feature values. Data labelling which is done by human who can cause error could be one of the triggers that can induce label noise as human tends to be bias and there might be element of subjectivity included during labelling. According to study proposed in (Koziarski et al., 2020), malicious attack as in data poisoning (Li et al., 2016) where attackers intentionally modify the class label could corrupt the data and consequently yield label noise.

Insufficient useful information is one of the factors that can lead to noisy environment (Nigam et al., 2020) besides misconceptions during data labelling (Koziarski et al., 2020). Noise with minority class label which located inside the majority class region should be given more attention than noise examples with majority class label within minority region. This is due to the fact that the minority class itself does not provide

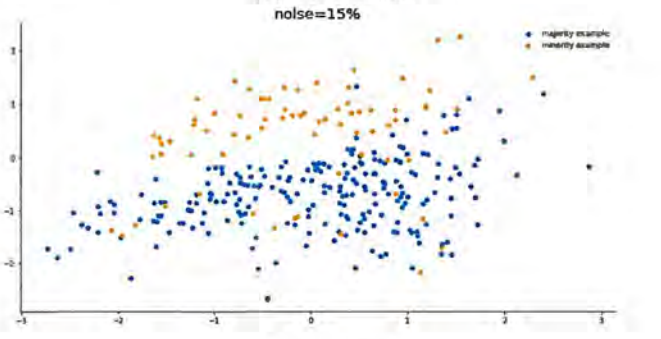
enough information and its examples are prone to be misclassified during classification as suggested by (Ju et al., 2021). Aggravation of complexity of induced models might happen due to noisy environment where the classifier will face difficulties in distinguishing the class label of an example and in determining the complex decision boundary thus prolonging the training running time (Pelletier et al., 2017).



(a) 5% noise.



(b) 10% noise.



(c) 15% noise.

Figure 2.1. Imbalanced data affected by different levels of label noise; 0.05, 0.10, 0.15. Such noisy environment hinders the predictive performance of a classifier due to insufficient information related to minority class which eventually will make the significance of this class is reduced while increasing the misclassification rate.

2.2.2 Class Overlap/Separability

Class overlap or class separability is one of the major problems responsible for degradation of classifier's performance. It refers to examples that belong to different classes are found in an ambiguous region due to the closeness in their feature values (Nekooeimehr and Lai-Yuen, 2016; Vuttipittayamongkol et al., 2020) which can be observed in Figure 2.2. Study conducted in (Sasada et al., 2020; Fernandes and de Carvalho, 2019) suggest most standard classifiers manage to produce an acceptable predictive accuracy rate despite the value of imbalance ratio of datasets. Therefore, it can be observed that imbalance ratio is not the only factor needed to be looked upon when addressing imbalanced classification problem but also class overlap since it plays a weighty influence on deciding the accuracy of classification (Napierala and Stefanowski, 2016; Yuan et al., 2020; Vorraboot et al., 2015).

Region around decision boundary usually contains examples from both minority and majority classes where the distribution of minority examples in this overlapping region is sparser and less dense compared to majority class. Hence, minority examples in this region could be easily mistaken as majority example as they are outnumbered by majority examples. However, generating more minority examples within borderline region or overlapping region could tackle this class overlap problem and enhance the classification of minority class (Vuttipittayamongkol and Elyan, 2020).

2.2.3 Small Disjuncts

The presence of small-sized clusters containing examples from one class and located in the in the region of opposite class due to underrepresented subconcepts in imbalanced problem is known as small disjuncts (Almutairi and Janicki, 2020) as presented in Figure 2.3. Within-class imbalance which corresponds to this small disjuncts problem is yet to be solved in the recent research works as most of them manage to alleviate

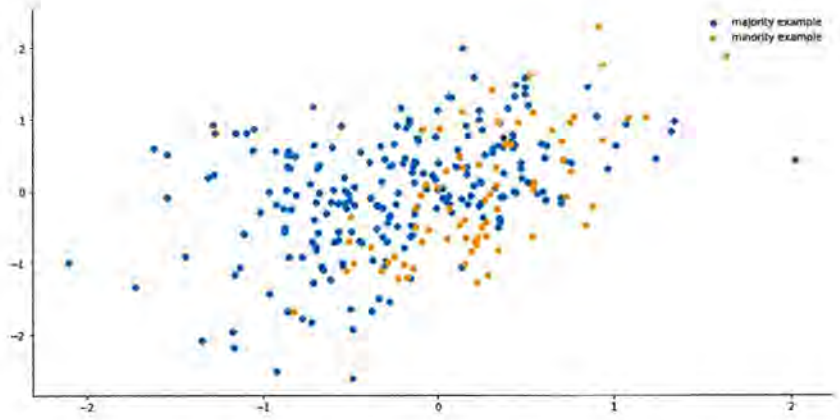


Figure 2.2. The class overlapping between minority and majority examples where in the decision boundary cannot be figured out clearly. This problem could lead to an increment in the classification complexity.

only the between-class imbalance. (García et al., 2015; Leevy et al., 2018; Cheng et al., 2019b) imply that small disjuncts problem is closely related to noise as an increase in the number of noise examples could induce more small disjuncts in the dataset. Most classifier models have difficulty in covering minority class that has been divided into several sub-concepts, leading to lower accuracy of identifying minority examples.

Clustering-based resampling method is one of the solutions proposed in solving within-class imbalance besides detection of small disjuncts where the data is grouped, forming clusters and then performs resampling in the clusters either generating valuable examples or removing some useless examples (Cieslak et al., 2006; Santos et al., 2015; Lin et al., 2017; Douzas et al., 2018).

2.2.4 Complexity Measures in Imbalanced Dataset

The obvious complexity measure available for imbalanced class dataset is to compute the ratio between minority class distribution and majority class distribution. It is commonly known as Imbalance Ratio (IR). Equation 2.1 gives the IR definition (Ab-

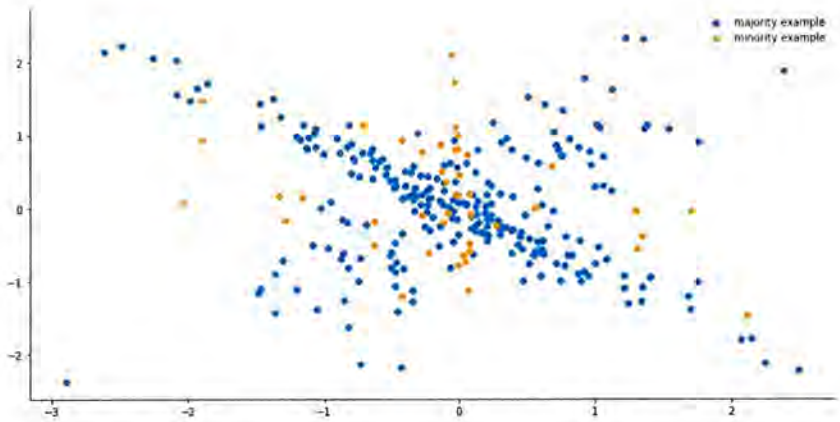


Figure 2.3. Sub-clusters of minority class are located inside majority class region which could give arise to small disjuncts problem.

dulhammed et al., 2019; Kovács, 2019a):

$$\text{IR} = \frac{\text{total samples in majority class}}{\text{total samples in minority class}} \tag{2.1}$$

Regardless, there are other various other complexity measures have been proposed to compliment IR values. In (Ho and Basu, 2002), a Fischer's discriminant (F1) ratio has been proposed to measure the means and variance of dataset classes. F1 ratio is one of the feature overlapping measures that analyse the predictive variables' ability to discriminate. The majority of the measures examine the features alone and choose the most distinguishing feature, while some combine the individual feature evaluations (Barella et al., 2018, 2021). Equation 2.2 defined the F1 ratio.

$$f = \frac{(\mu_{c_i} - \mu_{c_j})^2}{\sigma_{c_i}^2 + \sigma_{c_j}^2} \tag{2.2}$$

In Equation 2.2, μ_{c_i} and μ_{c_j} is the mean of feature values that associated with class c_i and class c_j respectively, while $\sigma_{c_i}^2$ and $\sigma_{c_j}^2$ is the variance of features values that

associated with class c_i and c_j respectively. F1 outputs the maximum f among all input features. The higher the F1 value, the simpler is the classification problem concerning feature separability, and vice versa.

Barella et al. (2018) introduces neighbourhood-based complexity measures known as N1, N2 and N3. The Nearest Neighbour (NN) notion is used in neighbourhood measurements for evaluating classification challenges. For instance, they analyse the form of decision borders and class distributions using the distance between instances.

N1 uses Minimum Spanning Tree (MST) that connects all the examples from the dataset based on their distance, regardless of which class the example belongs to. The idea is to identify at least two examples from two different classes that are close to each other. It is assumed that those examples exists in the borderline of two classes, and the frequency of those examples with relative to all examples in the dataset is defined as $N1$. Thus, the complexity measure $N1$ is defined such as follows:

$$N1 = \frac{1}{n}\sum_{i=1}^{n} I((\mathbf{x_i}, \mathbf{x_j}) \in \text{MST} \wedge y_i \neq y_j) \tag{2.3}$$

where $I(\mathbf{x_i}, \mathbf{x_j})$ represents a connection between instances $\mathbf{x_i}$ and $\mathbf{x_j}$ and MST represents the set of all connections in the tree. Higher N1 value indicates the need for more complex boundaries to separate the classes and/or that there is a large amount of overlapping between the classes.

While N1 is known as the fraction of points on the class boundary, N2 is the ratio of average intra/inter class NN distance. The class distribution within each class and between classes is distinguished by N2. It is calculated for each instance how far it is from both the NN of the same class (intraclass) and the NN of a different class (interclass). The intraclass distance average to the interclass distance average ratio is

called N2.

On the other hand, N3 is defined as leave-one-out error rate of the 1NN classifier. The leave-one-out error rate of the 1-nearest neighbor (1-NN) classifier refers to the error rate obtained when using the leave-one-out cross-validation technique with the 1-NN algorithm. In leave-one-out cross-validation, each data point in the dataset is taken as a test sample, and the remaining data points are used as the training set. For each test sample, the 1-NN classifier identifies the nearest neighbor in the training set and assigns the class label of that neighbor to the test sample. The error rate is then calculated based on the misclassified test samples. Greater values of N2 and N3 signify more complex challenges (Barella et al., 2021).

This study decided to employ N1 in the analysis because it has a greater impact on the prediction results of specific classifiers such as SVM, *k*-NN, Decision Tree and ANN (Lorena et al., 2019) whereby most of the mentioned classifiers are used in the experimentation to develop the classification model followed by forecasting the class label. Apart from that,Lorena et al. (2019) points out that N1 is among the highest scoring meta-features that provide insight into how well algorithms perform when applied to the data by Random Forest, as they take into account neighbourhood information derived from the data.

Another approach in Datta et al. (2017) is to consider the existence of small disjuncts in the imbalanced class dataset, known as Geometric Small Disjuncts Index (GSDI). Once the small disjuncts are identified in dataset, the GSDI will measure the complexity of those disjuncts based on formulation defined in Equation 2.4.

$$GSDI = \left(\prod_{c=1}^{C} \frac{\sum_{i=1}^{\delta_c} \exp(-|\Delta_i^c|) \times \frac{m_c^{i'}}{m_c^i}}{\sum_{i=1}^{\delta_c} \exp(-|\Delta_i^c|)} \right)^{\frac{1}{C}} \tag{2.4}$$

In Equation 2.4, δ_c is the number of disjuncts within the c-th class. The Δ_i^c is the i-th disjunct within the c-th class containing m_c^i test points, $m_c^{i'}$ out which are correctly classified. Higher GSDI indicates the identified small disjuncts in particular dataset will help in reducing the learning complexity, while lower GSDI gives arise to the learning complexity.

2.3 SMOTE: Synthetic Minority Oversampling Technique

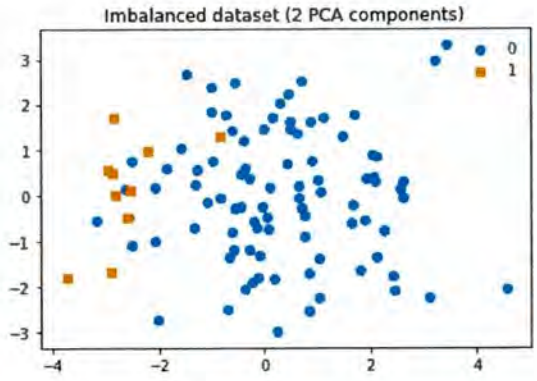
There are several ways of handling imbalanced class datasets, which can be categorized into two methods; data level and algorithm level. Algorithm level method enhances a classifier's accuracy on the minority class by tuning the existing classification algorithm. For example, aiming to highlight any misclassification or correct classification in the positive (minority) class, the cost imposed on the false negatives is larger than the false positives. This paper focuses on the data level method which uses preprocessing steps in rebalancing the class distribution by either implementing oversampling or undersampling to decrease the imbalance ratio (IR) of data-sets.

Oversampling techniques are defined as replicating minority examples randomly to increase the minority class size or new examples are generated synthetically from the interpolation of minority examples respectively. Whereas in the undersampling technique, the process occurs in the majority class wherein the examples are eliminated blindly or by using a heuristic approach to balance the number of examples in both classes (Spelmen and Porkodi, 2018). However, both oversampling and undersampling have their own drawbacks where replicating examples could lead to over-fitting as it does not give additional information about minority class while examples reduction in majority class could cause the loss of potentially valuable information especially in the case of small-sized datasets (Douzas et al., 2018; Le et al., 2019; Sowah et al., 2016).

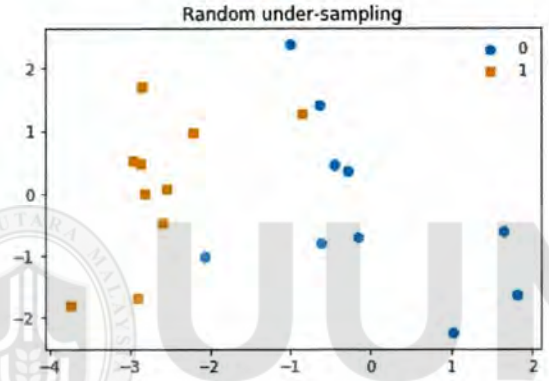
In the case of large-sized imbalanced datasets, the undersampling method performance is quite effective and convenient because the information loss is insignificant due to downsizing the size of majority samples (Maldonado et al., 2019; Guzmán-Ponce et al., 2020). Despite all that, it has been observed that there is a less risk when applying the oversampling method to real-world datasets compared to undersampling (Cheng et al., 2019a; Douzas and Bacao, 2017). As a result, the oversampling method has gained more attentions from researchers, resulting in many variants created to make sure that misclassification of minority class samples in imbalanced datasets problems can be reduced, thus enhancing the accuracy of classification (Mullick et al., 2019; Zhu et al., 2017; Bennin et al., 2017; Wang et al., 2019). Figure 2.4 illustrates the resulting data after performing undersampling and oversampling process, over imbalanced dataset.

Synthetic Minority Oversampling Technique (SMOTE) was introduced by (Chawla et al., 2002) to mitigate the overfitting problem occurring from random oversampling (Kovács, 2019b; Maldonado et al., 2019; Liang et al., 2020) and (Kaur and Gosain, 2018) shows that SMOTE has good robustness in noisy environment compared to random oversampling. SMOTE, as described in Algorithm 1, adds synthetic data using k-nearest neighbour algorithm by calculating the Euclidean distance and chooses the nearest neighbours for a given example from the neighbourhood provided as an input. The synthetic data is created by identifying the density of examples clustered in minority class so that SMOTE can find the actual examples which will give valuable information about the minority class labels and outliers that might influence the outcome gravely. The description of SMOTE can concisely be expressed in the following form (Santoso et al., 2017):

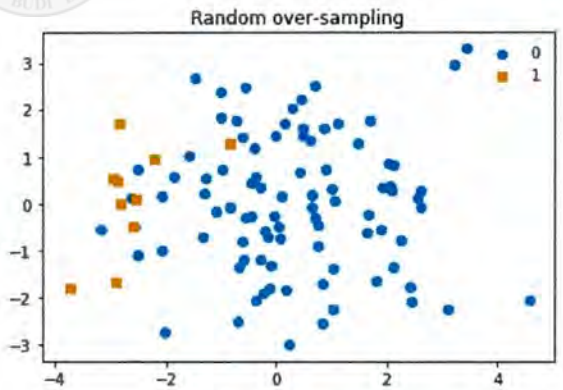
$$D_{new} = D_i + (D_l - D_i) \times w \tag{2.5}$$



(a) Original data.



(b) Undersampled data.



(c) Oversampled data.

Figure 2.4. Three figures show the difference between (a) original data, (b) undersampled data, where data are removed from the dataset, and (c) oversampled data, where duplicate data are added to the dataset, which can be performed to solve imbalanced classification problem.

Where D_{new} = synthetic data, D_i = examples from minority, D_l = one of k-nearest neighbour from D_i, w = random number between 0 and 1.

Algorithm 1 SMOTE

Input:
$\mathbf{X}_+ \leftarrow [x_1, x_2, \ldots x_n], x \in \mathbb{R}^m, m \in \mathbb{Z}^+$ ▷ Minority class samples
$T \leftarrow n$ ▷ Number of minority class samples
$N \leftarrow$ *Amount of SMOTE*
$k \leftarrow$ *Number of nearest neighbors*
Output: *(N/100) * T synthetic minority class samples*

procedure SMOTE PROCEDURE
  if $N < 100$ **then**
    $\mathbf{X}_+ \leftarrow Permutate(\mathbf{X}_+)$ ▷ Permutate the order of $\mathbf{X}_+$
    $T \leftarrow (N/100) \times T$
    $N \leftarrow 100$
  $N \leftarrow (int)(N/100)$ ▷ Assumed to be integral multiples of 100.
  $\mathbf{X}'_+ \leftarrow [x_1, x_2, \ldots x_T] \in \mathbf{X}_+$ ▷ T minority class samples
  $\mathbf{D} \leftarrow \emptyset$ ▷ Initialize empty set of distance matrix
  $\mathbf{X}'' \leftarrow \emptyset$ ▷ Initialize empty set of selected samples
  $\mathbf{S} \leftarrow \emptyset$ ▷ Initialize empty set of synthetic samples

  ▷ Computing the nearest neighbour of each selected sample
  for $x_i \in \mathbf{X}'_+$ **do**
    for $x_j \in \mathbf{X}'_+$ and $i \neq j$ **do**

      ▷ Computing the distance between x_i and x_j, where $p = 2$
      $\mathbf{D}_{i,j} \leftarrow \left(\sum_{k=1}^{m} |x_i^k - x_j^k|^p\right)^{\frac{1}{p}}$
  for $x_i \in \mathbf{X}'_+$ **do**
    for $g \leftarrow 1$ to k **do**
      $\mathbf{X}'' \leftarrow x_j, argmin_j(D_i)$ ▷ Select x_j with closest distance to x_i
      $\mathbf{D}_{i,j} \leftarrow \emptyset$ ▷ To prevent reselecting the same sample again

  ▷ Generating synthetic data
  for $x_i \in \mathbf{X}'_+$ **do**
    $N' \leftarrow N$
    $q \leftarrow Randomize(1, k)$ ▷ A random integer between 1 and k
    while $N' \neq 0$ **do**
      $gap \leftarrow Randomize(0, 1)$ ▷ A vector of random number
      $\triangle x = x_i - \mathbf{X}''[i \times q + q]$ ▷ Compute a difference vector
      $s \leftarrow x_i + gap \odot \triangle x$ ▷ A synthetic sample is generated
      $\mathbf{S}.append(s)$ ▷ Populating the set $\mathbf{S}$
      N'--

SMOTE obviously could remove class imbalance problem through generation of synthetic examples without involving removal of existing examples which can cause information loss. Moreover, linear interpolation between selected minority examples may prevent redundant and replicated examples from being produced as in random oversampling algorithm (Verbiest, 2014). Therefore, SMOTE successfully overcomes over-fitting problems and enhances the learning task of most classifiers.

Regardless, there is still much to be desired from the conventional SMOTE. For the starter, one of its drawbacks is that SMOTE does not have a mechanism to control

where the synthetic example should be placed in the first place. It is not uncommon that new synthetic examples to appear at an inappropriate position in the hyperspace especially in the opposite region because the noisy information is being distributed (Guan et al., 2020) as shown in Figure 2.5. Noise examples extension will eventually lead to over-generalization problem. Apart from that, SMOTE exacerbates class overlap problem by adding more overlapping examples as it cannot differentiate overlapping regions from so-called safe regions (Cheng et al., 2019a). Next, SMOTE has the possibility of generating useless synthetic examples as well as only able to tackle between-class imbalance while setting aside the small disjuncts or within-class imbalance. Taking within-class imbalance into account can give quite an impact when trying to classify minority examples more accurately (Douzas et al., 2018).

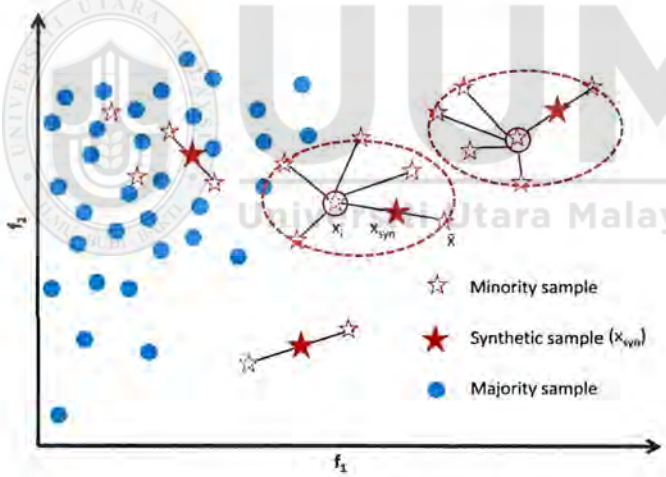


Figure 2.5. SMOTE linearly interpolates a randomly chosen minority class example and one of its k=5 nearest neighbors. The presence of noise might cause minority examples generated in the majority class region.

2.3.1 Overview of SMOTE Variants Strategies

Before diving into SMOTE discussion, this section will overview some common strategies that are used in order to improve the effectiveness of SMOTE. The excellent performance shown by SMOTE algorithm has motivated the appearance of many variants recently either through modification of SMOTE or extension of SMOTE (Soltanzadeh and Hashemzadeh, 2021). The former considers the characteristics of examples

when changing the region of oversampling whereas the latter refers to combination of SMOTE algorithm either with pre-processing or post-processing approaches by maintaining the basic algorithm.

Modification of SMOTE is normally associated with change-direction algorithm where it attempts to redirect the synthetic examples into certain preferred areas so that it can control the distribution of these examples. Techniques proposed in (Han et al., 2005; Bunkhumpornpat et al., 2009; Liang et al., 2020; Kunakorntum et al., 2020; Chen et al., 2020) are of change-direction-based SMOTE variants which produce synthetic examples as close as possible to specific regions and most of these techniques prefer regions with high concentration of minority examples.

Filtering, clustering and weighting are the most common techniques integrated with SMOTE algorithm in extension of SMOTE. Different types of filtering techniques are implemented in (Batista et al., 2004; Ramentol et al., 2012; Sáez et al., 2015; Cheng et al., 2019a) for data cleaning purpose by detecting and removing either noise examples or overlap examples. Some SMOTE extensions (Barua et al., 2012; Nekooeimehr and Lai-Yuen, 2016; Douzas et al., 2018; Wei et al., 2020a) suggested clustering and weighting algorithm are best to be applied together while other technique (Ramentol et al., 2012) only combined SMOTE with clustering technique. Clustering algorithm could guarantee that generation of synthetic examples occurs within minority class region and weighting algorithm is applied to determine which minority examples are appropriate and suitable for oversampling to produce useful information. Both algorithm are frequently found in SMOTE extensions that try to handle data with class overlap or small disjuncts problems.

2.3.2 Mitigating Noise Examples Problem

In the application of SMOTE algorithm, the presence of noise examples in imbalanced datasets can aggravate the noisy environment and poorly influence the performance of classifiers. Instead of generating synthetic examples in the minority region, these examples are deeply introduced in the majority region and this phenomenon is called over-generalization. Thus, it is important to remove the noise examples prior to oversampling or to filter them posterior to oversampling.

SMOTE-Edited Nearest Neighbour Rule (SMOTE-ENN) is a hybrid sampling technique which combines SMOTE and data cleaning (Batista et al., 2004). The motivation behind this technique is to discard noise examples both from minority and majority classes using ENN after application of SMOTE on the training set (Yan et al., 2019). ENN detects noise examples by randomly choosing any examples and considers their three nearest neighbours (Douzas and Bacao, 2019). Therefore, the noise examples that are generated during SMOTE can be deleted by applying ENN afterward in order to reduce noisy environment that will confuse the classifier. Even though SMOTE-ENN is able to remove noise examples using data cleaning approach, problem might happen when the minority class has sparse distribution and small in size since the hard-to-learn minority examples overlapping with majority examples tends to be removed by this technique. This phenomena can easily downgrade the generalization of underpresented minority class due to compression of class space (Guan et al., 2020).

Modification of SMOTE like Borderline-SMOTE (Han et al., 2005) does not employ noise removal, instead it does ignore noise examples and not include them in the oversampling process. This technique takes into account the majority examples distribution and makes assumption that examples existing close to borderline have more contribution to classification success (Yun et al., 2016). As minority examples are grouped

into three categories which are safe, borderline and noise based on the number of minority and majority examples in their neighbourhood prior to oversampling, any noise examples that exist in minority class are filtered effectively (Al Majzoub et al., 2020). Then, only examples in the borderline region are used for oversampling as these examples are easily misclassified during classification. It can be said that noise filtering also occurs in majority class as noise minority examples are being excluded. B-SMOTE1 and B-SMOTE2 are two versions of Borderline-SMOTE in which the synthetic examples are created between the borderline examples from minority class and their nearest neighbours from the same class in the former while the latter generates synthetic examples by adding majority class examples in the interpolation of a minority example. The downside of this technique is that an example can only be identified as noise if all of its nearest neighbours are from majority class. If there are two minority examples placed closely to each other within the majority class region, they are not considered as noise since one of the nearest neighbours is of the same class (Piri et al., 2018).

ENN filtering algorithm might be suffering from failure of detecting all noise examples that are identified based on the distance of nearest neighbours. To improve the solution, authors in (Sáez et al., 2015) proposed combination of SMOTE with another noise filter known as Iterative-Partitioning Filter (IPF) which depends on classifiers for identification of noise examples, instead of using distance metric just like SMOTE-ENN. Noise examples introduced in the data as well as the existing ones from both majority and minority classes can be removed easily. In addition, this technique is effective in addressing the problems caused by noise and borderline examples (Chen et al., 2016; Agrawal et al., 2017). The algorithm starts with dividing the oversampled training data into n equal-sized subsets where union of any $n-1$ subsets will be used independently for training multiple n classifiers. Noise identification is made with two voting schemes called consensus and majority. An example is considered as noisy if

all classifiers misclassify it in the consensus scheme whilst in majority scheme,if more than half of the classifiers do misclassification of the example. Then, the noise examples are added into a new set, S. For a number of consecutive iterations t, if the size of noise examples detected in each of the iterations lower than a $p\%$ of the original training dataset size, then the iterations will stop. Lastly, all the noise examples in set S are deleted from the training dataset before classification is further performed. Even though IPF relies on certain classifiers, it can face similar risk as ENN in handling small-sample imbalanced dataset (Guan et al., 2020) such as removing minority examples from the overlapping region where minority class has lower local distribution compared to majority class.

The purposes of Geometric SMOTE (GSMOTE) proposed by Douzas and Bacao (2019) are to generate a variety of non-noisy synthetic minority examples within a safe truncated hyper-spheroid region which provides new valuable information and eventually prevent over-fitting that is usually induced by redundant example. SMOTE algorithm applies minority selection strategy where the k nearest neighbours of a randomly chosen minority examples is chosen from the same class only. This strategy tends to encourage generation of noisy example in the majority region if one of the nearest neighbours is found deep in this region. This technique expands the selection phase by including majority and combined selection strategies which allows safe expansion of minority class area besides restricting the generation of noisy examples.

A grouped SMOTE algorithm with noise filtering mechanism (GSMOTE-NFM) (Cheng et al., 2019a) implements a different way of identifying and filtering noise examples by comparing the probability density of same example in two different classes. Unlike SMOTE-ENN and SMOTE-IPF which use distance between nearest neighbours and relies on the voting schemes of classifiers respectively. The prior distributions of both

majority and minority classes are learned respectively beforehand through construction of Gaussian-Mixture Model (GMM). After filtering noise examples, GMMs are adopted on the remainder data and the probability distribution information is used to split the minority class into three different groups; safe, borderline and outlier. Lastly, before SMOTE implementation, examples of each group is given an individual parameter k which depends on their characteristics of distributions. The demerits of GSMOTE-NFM are the construction of GMM is time-consuming and if the data has high imbalanced ratio, the extremely sparse minority examples will jeopardize the accuracy of GMM model (Chen et al., 2020).

SMOTE algorithm of limiting the radius (LR-SMOTE) (Liang et al., 2020) modifies the original SMOTE algorithm just like Borderline-SMOTE does but their difference can be observed from the objectives perspective. While Borderline-SMOTE aims to change the region of oversampling, LR-SMOTE goal is to set a limit range to the oversampling space since noise generation can happen due to boundless space for synthetic examples generation using SMOTE. Existing noise examples can worsen the noise generation as these examples uncontrollably invade the opposite class region (Liu et al., 2020). Therefore, LR-SMOTE performs denoising step by using SVM algorithm for classifying the training data to find misclassified minority examples. These examples are then placed into one sample set and their k number of minority nearest neighbours is observed for detecting and deleting noise examples. Then, K-means clustering algorithm (Zhang et al., 2018) is adopted to find the centre point of minority class for calculating the Euclidean distance, d between this point and every non-noise minority examples as well as the average of all distances, d_{mean}. Synthetic examples generation formula is described such in Equation 2.6:

$$D_{new} = D_i + (D_c - D_i) \times rand(0, M) \tag{2.6}$$

where D_{new} = synthetic data, D_i = examples from minority, D_c = minority class centre point, $rand(0,M)$ = random number between 0 and 1. For this technique, the range for generation of synthetic examples is changed into 0 and M where M is the ratio between d_{mean} and d thus synthetic examples are produced in the region closer to this point. The final step includes filtering the excessive synthetic examples to balance the size of dataset. Experimentation results have shown that the effectiveness LR-SMOTE algorithm is proportional to the imbalanced ratio of imbalanced dataset.

SMOTE-ENN and SMOTE-IPF are amongst SMOTE variants that implement particular noise filter and introduce additional parameters in the algorithm. On the other hand, self-adaptive robust SMOTE (RSMOTE) (Chen et al., 2020) is able to boost the performance of classifiers without applying these two steps. Noise examples in minority class region are identified and filtered out heuristically based on the number of its k nearest neighbours (Han et al., 2005) so that these examples will not be operated in the next step. Next, a new set is created to gather all the noise-free remaining minority examples, resulting in only safe and borderline minority examples are included in this set. In order to differentiate between these two type of minority examples, the ratio of the total distance of every minority examples with their nearest neighbours from the same class to the total distance of this example from the nearest opposite class examples, also known as relative density is calculated. 2-means clustering is deployed to divide these examples inside the set into two clusters, one with higher relative density of examples are residing inside it while the other has smaller relative density of examples. For each minority examples in both clusters, the number of majority examples in its nearest neighbours helps to define chaotic level which is used to reweigh the amount of synthetic examples needs to be generated. Safe examples are given higher weights to be oversampled so that fewer synthetic examples are placed near the borderline region. In this way, decision boundary will be more obvious besides the

overlapping can be avoided effectively.

In summary, there are two SMOTE variant approaches when considering mitigating the noise examples' problem. Noise examples can be filtered either prior to SMOTE application (pre-processing) or post SMOTE application (post-processing). Table 2.1 tabulates the SMOTE variants that fall into these two approaches.

Table 2.1
Two SMOTE variants' approaches in solving noise examples problem.

Approach	SMOTE Variants
Pre-processing	BD-SMOTE Han et al. (2005), SL-SMOTE Bunkhumpornpat et al. (2009), NRAS Rivera (2017), GSMOTE (Douzas and Bacao, 2019) ,GSMOTE-NFM Cheng et al. (2019a), LR-SMOTE Liang et al. (2020), R-SMOTE Chen et al. (2021)
Post-processing	SMOTE-ENN Batista et al. (2004), SMOTE-RSB* Ramentol et al. (2012), SMOTE-IPF Sáez et al. (2015), SMOTE-PSO Cervantes et al. (2017)

2.3.3 Mitigating Class Overlap/Separability Problem

The existing class overlap problem might be aggravated after SMOTE is applied to address the imbalance problem since it can introduce more examples in the overlapping region and make decision boundary becomes unclear. The importance of borderline examples that are located in the overlapping region is an issue that need to be given more attention because they can either be noise examples or valuable examples that contain information (Fernández et al., 2018). Thus, most of SMOTE variants which trying to overcome class overlap focus on the hard-to-learn examples located around class boundaries.

SMOTE-Tomek Links (SMOTE-TL) (Batista et al., 2004) alleviates the overlap class problem by removing the pairs of examples that belong to different classes. Tomek

links can be defined as pair of minimally Euclidean distanced neighbours of different classes and it can either be used to do under-sampling task or data cleaning task (Sanguanmak and Hanskunatai, 2016). In SMOTE-TL, synthetic examples are initially generated before Tomek Links are identified and later the distance between a synthetic examples and both examples in the pair are measured. The distance must be smaller than the distance of the example pair for elimination of the pair that is considered as overlapped, thus any examples that cause overlapping are removed effectively. Tomek links that are going to be eliminated mostly found in the borderline region due to overlapping tends to take place here. Nevertheless, noise generation cannot be successfully avoided as SMOTE-TL depends on the adjacencies between borderline examples (Sasada et al., 2020). Figure 2.6 presents how this method operates.

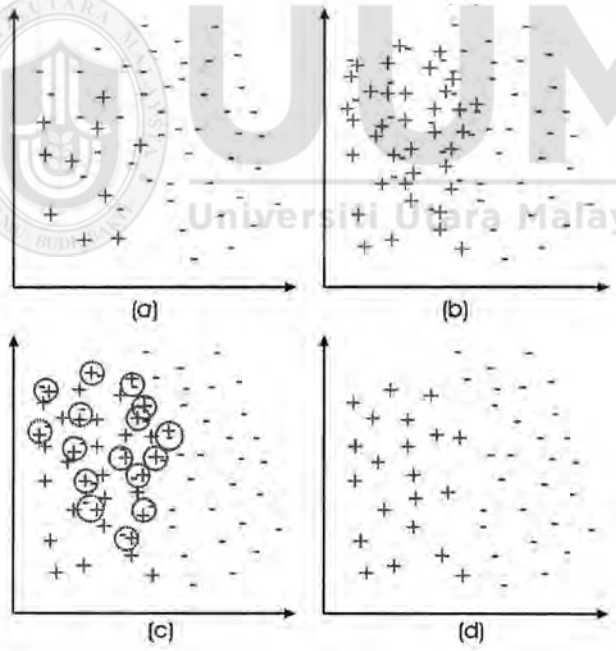


Figure 2.6. (a) presents an imbalanced dataset with little information regarding minority class. In (b) and (c), SMOTE-TL is implemented to generate more synthetic minority examples and pairs of overlapping examples from both minority and majority classes are detected respectively. These pairs are then deleted from the dataset, removing overlapping region that could increase the complexity of decision boundary.

Borderline-SMOTE is not only capable of tackling noisy environment but also class

overlap problem. This technique highlightes the importance of minority examples in the borderline region and these examples are chosen to be oversampled using linear interpolation which can increase the density of minority class in this region and subsequently might enhance the visibility of class border.

Clustering is one of the algorithms used to deal imbalanced datasets with class overlap, therefore Density-Based Synthetic Minority Over-sampling Technique (DBSMOTE) (Bunkhumpornpat et al., 2012) is proposed where the algorithm consists integration of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and SMOTE. Motivated by Borderline-SMOTE, this technique carefully oversamples minority examples in the borderline region to maintain the accuracy of majority class that declines in Borderline-SMOTE. DBSCAN algorithm (Schubert et al., 2017) is applied because of its ability to find cluster with an arbitrary shape by using the minimum density level estimation where the minimum number of neighbours, *minPts* and radius *eps* are the parameters.

Firstly, the algorithm constructs a new data structure called directly density-reachable graph (Bunkhumpornpat and Sinapiromsaran, 2017) which transformed from the DBSCAN algorithm and works as an underlying weighted directed graph. This graph contains core examples that have examples more than *minPts* within *eps* distance and borderline examples which lie in their neighbourhoods. It can be concluded that the latter are directly density-reachable to the former. The borderline example can be formed if at least one of a pair examples is directly density-reachable to others, that is how an arbitrarily shaped cluster can be discovered. Then, generation of synthetic example occurs along the shortest path between a pseudo-centroid of a minority class cluster and each example in that class. Subsequently, overlapping can be avoided as the synthetic example is located on the shortest path that is formed inside the minority

cluster. However, over-fitting still could not be solved due the density of synthetic examples are higher close to the cluster centroid (Wong et al., 2016).

Majority Weighted Minority Oversampling Technique (MWMOTE) (Barua et al., 2012) also employs clustering algorithm to make sure that all synthetic examples lie inside the minority class regions, such that it prevents them from overlapping with majority class regions (Wei et al., 2020b,c). This technique aims to improve the initial selection process besides producing informative and necessary synthetic examples from hard-to-learn minority examples as shown in Figure 2.7. Borderline-SMOTE tends to synthesize duplicated and meaningless examples, therefore this technique attempts to overcome the limitation.

Essentially, MWMOTE has three steps involved. Initially, it identifies the most difficult to learn minority class examples from the original set of minority class which can be found near to the borderline of both classes. Instead of applying the well-known k-NN algorithm, k-nearest neighbours are observed by using the euclidean distance in which the value of k is determined by the user. The optimal value of k is contingent upon the situation, and finding the ideal value could involve some experimentation and careful evaluation on the specific problem and dataset. (Ali et al., 2019b) evaluates the classifier performance to determine the optimum k for heterogeneous dataset. k-NN algorithm cannot find the hard-to-learn minority examples appropriately even though they are located within borderline region if these examples k-nearest neighbours are of the same class. Next, three factors are taken into consideration before assigning selection weight to every example in the set of identified examples: 1) the closeness of minority examples to majority class region, 2) the sparsity of minority class cluster and 3) the sparsity of majority class cluster. Lastly, this technique constructs M sub-clusters from S_{min} using modified hierarchical clustering algorithm prior to over-

sampling. This can prevent the synthetic examples from overlapping with the opposite class examples such that they all lie in the minority class cluster.

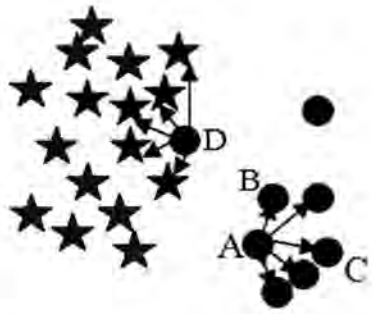


Figure 2.7. *A* and *B* which are placed near the borderline of majority class cannot be identified as hard-to-learn examples by using *k*-nearest neighbour (*k*-NN) since the nearest neighbours have the same class label. MWMOTE algorithm is able to detect these examples as hard-to-learn by implementing a different approach nearest neighbours. The identification of these examples is crucial in MWMOTE as it focuses on generating more synthetic examples nearby example that probably has more helpful information compared to the others.

Even though MWMOTE could identify borderline examples, small-sized sub-clusters containing significant information that located far away from the majority class can be easily undetected (Wei et al., 2020a). Therefore, Adaptive semi-unsupervised weighted oversampling (A-SUWO) (Nekooeimehr and Lai-Yuen, 2016) is introduced to overcome the above problem by implementing a semi-unsupervised hierarchical clustering approach that enables detection of small clusters. A-SUWO technique can be considered as an extension of MWMOTE as this technique also focuses on finding the hard-to-learn minority examples and performs oversampling based on the weights assigned to each minority examples in the sub-clusters while applying different clustering algorithm that adaptatively determines the size of synthetic examples in every sub-cluster.

Before clustering the minority examples, noise examples are identified and eliminated using Borderline-SMOTE (Han et al., 2005) method such that they will not be included in the next process. While MWMOTE only performs hierarchical clustering on

minority class, this technique performs the clustering on both minority and majority classes so that the number of synthetic minority examples overlapping with majority examples can be minimized . However, the hierarchical clustering is modified for application on minority class in order to observe the existence of majority sub-clusters. Considering two minority sub-clusters are going to be combined, if the Euclidean distance of a majority cluster to both of them less than the a distance threshold T, there will be no combination happening, preventing majority examples from being included in the cluster and that is how semi-unsupervised hierarchical clustering works in this technique. On the assumption that more synthetic examples needed in sub-clusters with higher misclassification error rate, cross validation is used to calculate the misclassification error of each minority sub-cluster or in other words, the classification complexity. Sub-clusters that have higher misclassification error will be given a bigger size compared to those with lower error rates since they need more information in the cluster that will increase the classification accuracy (Wei et al., 2020a).

After that, the probability distribution of every minority examples in each sub-clusters needed to be measured based on the weights before oversampling can be done. Instead of assigning weights on minority examples in each sub-cluster based on the distance to all candidate borderline majority examples as in MWMOTE, A-SUWO finds the average Euclidean distance to the k-nearest neighbours from majority class. This new approach gives more chances for minority examples that nearer to majority examples and have high misclassification rate to be oversampled. Apart from that, it performs weighting algorithm on all sub-clusters separately which result in small-sized sub-clusters are oversampled and not overlooked, contrary to MWMOTE that tends to ignore them. Then, generation of synthetic examples occurs only between minority examples from the same cluster where the initial selection of example is made by sampling from the pre-determined probability distribution, thus overlapping could be

avoided as all synthetic examples fall within the cluster. Despite being able to alleviate over-generalization due to weights assignment on hard-to-learn minority examples and also overlapping problem, this algorithm has a higher complexity which might lead to poor sampling (Wei et al., 2020c). In addition, over-fitting might even happen when synthetic examples are concentrated near the borderline regions, encouraging duplication of examples (Wei et al., 2020a).

Most of the listed SMOTE extension such as Borderline-SMOTE, DBSMOTE, MWMOTE, and A-SUWO address the class overlap problem through initial selection either aiming to enhance the separability of examples from different class or to increase the density of synthetic examples near the decision boundary. Contrary to those methods, SMOTE-IPF (Sáez et al., 2015) applies iterative filtering approach to handle the aforementioned problem by not filtering out the sparse minority examples in the overlapping region that usually assumed to be noise, instead eliminating more majority examples. This technique prones to remove majority examples from overlapping region in some type of datasets which causes the density of minority examples becomes higher. Therefore, the visibility of minority class becomes clear and the decision boundary is more regular (Vuttipittayamongkol and Elyan, 2020) since class overlap is removed.

MWMOTE and A-SUWO are not the only SMOTE variants that take the distribution of classes into consideration to alleviate class overlap problem. A new technique called a synthetic minority based on probabilistic distribution (SyMProD) oversampling (Kunakorntum et al., 2020) also applies the same concept. The algorithm starts with implementation of Z-score normalization on both majority and minority classes and the Z-score value is used for comparison with a noise threshold value (NT), to identify any noise examples exist in the original data which are then not included in

the training data. Next, rather than calculating the probability distribution according to weight (Nekooeimehr and Lai-Yuen, 2016; Wei et al., 2020a), this technique converts the ratio of closeness factor derived from both classes. Minority examples that will be selected for oversampling based on the probability distribution possibly can reduce the overlapping problem. This is due to class closeness factor is compared to the cutoff threshold (CT), which helps to only choose minority examples located in the minority class region, ignoring the examples found in the opposite region. The number of selected minority examples follows the difference between majority and minority examples in the original dataset. Lastly, the generation of synthetic examples step involves searching for M minority nearest neighbours of the referenced minority example. The real value of synthetic example will be generated within the range of real value of the nearest neighbours, ensuring the example is not duplicated and over-generalization could be hindered. Apart from that, there will be no overlapped synthetic examples generated since they are found in the minority distribution.

Even though LR-SMOTE (Liang et al., 2020) and RSMOTE (Chen et al., 2020) are direction-based algorithm, they still manage to handle class overlap problem in some way without identifying the hard-to-learn examples. generating more synthetic examples near the centre point of minority class or around safe region reduce addition of overlapping examples thus avoiding the problem from being exacerbated.

In conclusion, there are two main approaches in dealing with class overlap and class separability problem. The first approach is based on a "generate-to-eliminate" examples in the class border region. SMOTE-TL and SMOTE-IPF fall into this approach. The second approach is based on a "generate-to-increase" examples in the class border region so the visibility of the class border is increasingly clear to learning algorithm. The SMOTE variants that fall into this approach are BD-SMOTE, SL-SMOTE,

MSYN, DBSMOTE, MWMOTE, A-SUWO, IA-SUWO, SyMProD, LR-SMOTE, and RSMOTE.

2.3.4 Mitigating Small Disjuncts Problem

Existence of small-sized clusters or sub-concepts in imbalanced datasets is easily overlooked by SMOTE therefore this technique fail to address within-class imbalance while being able to solve between-class imbalance. To make it worse, small disjuncts encourage the degradation of classifiers since they are easily overlooked due to their sparse distribution and usually being mistaken as noise. Apart from that, small disjuncts might increase noisy environment if they are considered for oversampling process.

Cieslak et al. (2006) proposed Cluster-SMOTE which integrates k-means clustering with SMOTE where it applies SMOTE oversampling in every k clusters of minority class after dividing the class using k-means clustering. Cluster-SMOTE manages to handle within-class imbalance only to some extent (Wei et al., 2020a) since the number of synthetic examples in each cluster is not considered besides it has difficulty in determining the optimal number of clusters. Apart from that, Douzas et al. (2018) claims that this technique fails at removing small disjuncts problem.

Small disjuncts might induce noise examples when example from one small disjunct is identified as nearest neighbour of seed example from other disjuncts during oversampling due to inappropriate k value of nearest neighbours. Hence, AND-SMOTE (Yun et al., 2016) instinctively determines the optimal k-values for every minority examples as different examples required different number of nearest neighbours used in SMOTE depending on the characteristics of data. Minority example and its nearest neighbours are contained in a rectangular region and then this region is expanded and accumu-

lated towards the next nearest neighbour. The change of precision values based on the size of nearest neighbour in every iteration helps to identify the optimal k-value by choosing the drop point precision. Instead of using random number in the range [0, 1] to generate examples on the lines between two points, this technique uses a vector of random number in the same range. The difference between Cluster-SMOTE and AND-SMOTE is obvious where the former has the ability to lift within-class imbalance but cannot eliminate small disjuncts where as the latter avoids noisy environment caused by small disjuncts but does not attempt to solve within-class imbalance.

KMeans-SMOTE (Douzas et al., 2018) applies k-means clustering together with SMOTE to tackle class imbalance problem as well as noisy environment issue. In (Cieslak et al., 2006), clustering algorithm is implemented only on minority class where as in KMeans-SMOTE, clusters are formed for both minority and majority classes. In this cluster, the number of minority and majority examples will be used to calculate the imbalance ratio of the cluster which then determines either this cluster is suitable for oversampling or not. Then, KMeans-SMOTE observes the distribution of minority examples in the filtered cluster by measuring its density and sparsity so that it can assign appropriate number of synthetic examples to be generated for each clusters before implementing oversampling (Zheng, 2020). This is how within-class imbalance is tackled as this technique does not blindly generate the same amount of synthetic examples in every cluster. Apart from that, KMeans-SMOTE properly prevents introducing more noise examples and overfitting problem by generating informative examples inside the minority class cluster (Gao et al., 2020b). Nonetheless, this technique still faces the same limitations as Cluster-SMOTE where the optimal number of clusters yet to be looked upon and clustering algorithm used does not applicable when the clusters have irregular shape.

Despite the fact that A-SUWO (Nekooeimehr and Lai-Yuen, 2016) and IA-SUWO (Wei et al., 2020a) alleviate within-class imbalance due to their high capability of detecting minority class sub-concepts located far from decision boundary, it is still not clear whether small disjuncts problem are being taken care of.

In summary, cluster-based oversampling techniques are widely practiced by researchers when the data contains small disjuncts problem that degrades the classification performance gravely. Observing the density of a cluster and then the sparsity of examples inside it are very crucial in determining the amount of synthetic examples needed by the cluster. Because of that, the synthetic examples are distributed appropriately between the clusters and eventually solves within-class imbalance.

2.4 Evaluation Metrics for Imbalanced Dataset

Evaluation metrics in machine learning are quantitative measures used to evaluate the performance of a machine learning model. These metrics are used to assess the accuracy and effectiveness of the model's predictions and to compare the performance of different models.

Confusion matrix as shown in Table 2.2 is used to evaluate the performance of classifier where the True Positive (TP) and True Negative (TN) are the numbers of correctly classified positive (minority) and negative (majority) class examples, respectively. False Positive (FP) is the number of negative class examples misclassified as positive, and False Negative (FN) is the number of positive class examples misclassified as negative class. The confusion matrix is used to derive evaluation metrics like precision, recall and specificity which are defined as Equation 2.7, Equation 2.8 and Equation 2.9 respectively.

(Arafa et al., 2022) states that True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) confusion matrix parameters as shown in Table 2.2 are the primary parameters from which other performance metrics, such as Precision, Recall, and Specificity are computed in machine learning classification tasks defined as Equation 2.7, Equation 2.8 and Equation 2.9 respectively. Another statistic used to assess classification models with imbalanced datasets is the Matthew's Correlation Coefficient (MCC), which shows how much the predictive model outperforms a random guess (Jiao and Du, 2016). A metric called Cohesion's Kappa gauges the degree of congruence between expected and actual classes (Grandini et al., 2022). The Geometric-Mean (G-Mean) score, which is derived from the product of sensitivity and recall, is another significant performance indicator for imbalanced classification (Al Helal et al., 2016). G-Mean takes into account the proportion of examples that are correctly classified in both minority and majority classes, whereas F1-score is more concerned with the average performance of precision and recall (Yuan et al., 2020).

Table 2.2

Description on the confusion matrix.

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

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{2.7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{2.8}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}} \tag{2.9}$$

The choice of evaluation metrics depends on the type of problem being addressed and the goals of the model. For example, if the goal is to minimize false positives, precision might be the most important metric. If the goal is to minimize false negatives, recall might be the most important metric.

Standard classification metrics like accuracy may not be the best approach to assess a model's performance when dealing with unbalanced datasets. This is due to the fact that the model can reach high accuracy by consistently forecasting the majority class while utterly disregarding the minority class.

Instead, the following metrics can be used to evaluate the performance of a model on imbalanced datasets:

- F1-score: The F1-score is a commonly used metric for binary classification problems that combines precision and recall into a single value. It is the harmonic mean of precision and recall and ranges from 0 to 1, with a value of 1 indicating perfect precision and recall.
- Geometric mean (G-Mean): The G-Mean is a performance metric that is commonly used to evaluate imbalanced binary classification problems. It is the square root of the product of the sensitivity and specificity of the model, and ranges from 0 to 1, with a value of 1 indicating perfect classification performance.
- Matthews Correlation Coefficient (MCC): The MCC is a correlation coefficient between the observed and predicted binary classifications. It takes into account true and false positives and negatives and ranges from -1 to +1, with a value of +1 indicating perfect prediction.
- Cohen's kappa statistic: Cohen's kappa is a measure of inter-annotator agree-

ment that is often used in natural language processing and image recognition tasks. It measures the agreement between two annotators who classify samples into a finite number of categories. In machine learning, Cohen's kappa is often used to measure the agreement between human annotators and a machine learning model.

- Area under the receiver operating characteristic curve (AUC-ROC): The AUC-ROC is a performance metric used to evaluate binary classification models. It measures the ability of a model to distinguish between positive and negative samples across different thresholds. The AUC-ROC ranges from 0 to 1, with a value of 0.5 indicating that the model performs no better than random chance, and a value of 1 indicating perfect classification performance.
- Area under the precision-recall curve (PR AUC): The PR AUC is another performance metric used to evaluate binary classification models. It is similar to the AUC-ROC, but is based on the precision-recall curve, which plots the precision and recall of a model as the classification threshold is varied. The PR AUC measures the trade-off between precision and recall, and ranges from 0 to 1, with a value of 1 indicating perfect classification performance.

p_o is the observed proportion of agreement between the two annotators or between the human annotator and the machine learning model, and p_e is the expected proportion of agreement under random chance.

2.5 Summary

Various algorithms have been intentionally combined with the original SMOTE algorithm to mitigate the aforementioned weaknesses as well as enhancing the effectiveness of generation of synthetic examples. Table 2.3 shows SMOTE variants can be categorized into groups based on the algorithm implemented in the method and these

variants also are capable of tackling one or more previously mentioned data complexities. SMOTE variants are developed with multipurpose objectives such as alleviating the data complexities originally co-exist in the data or those that have been intensified by limitations of SMOTE.

Table 2.3

Classification of SMOTE variants (chronologically ordered), with algorithm and data complexity categorization. The following are the abbreviations used for describing different variants: Filtering-based - F, change-direction-based - CD, clustering-based - C, clustering+weighting-based - CW, noise pre-processing - Pre, noise post-processing - Post, "generate-to-eliminate" examples in the class border region - GTE, "generate-to-increase" examples in the class border region - GTI.

SMOTE Variant	Algorithm	Approach	Data Complexities		
			Noise	Class Overlap	Small Disjuncts
SMOTE-TL (Batista et al., 2004)	F	GTE		/	
SMOTE-ENN (Batista et al., 2004)	F	Post	/		
BD-SMOTE (Han et al., 2005)	CD	Pre,GTI	/	/	
CL-SMOTE (Cieslak et al., 2006)	C	-			/
SL-SMOTE (Bunkhumpornpat et al., 2009)	CD	Pre,GTI	/	/	
DBSMOTE (Bunkhumpornpat et al., 2012)	C	GTI		/	
SMOTE-RSB* (Ramentol et al., 2012)	F	Post	/		
MWMOTE (Barua et al., 2012)	CW	GTI		/	
SMOTE-IPF (Sáez et al., 2015)	F	Post,GTE	/	/	
AND-SMOTE (Yun et al., 2016)	CD	-			/
A-SUWO (Nekooeimehr and Lai-Yuen, 2016)	CW	GTI		/	
NRAS (Rivera, 2017)	CD	Pre	/		
SMOTE-PSO (Cervantes et al., 2017)	F	Post	/		
KM-SMOTE (Douzas et al., 2018)	CW	-		/	/
GSMOTE-NFM (Cheng et al., 2019a)	F	Pre	/		
LR-SMOTE (Liang et al., 2020)	CD	Pre,GTI	/	/	
IA-SUWO (Wei et al., 2020a)	CW	GTI		/	
SyMProD (Kunakorntum et al., 2020)	CD	GTI	/	/	
RSMOTE (Chen et al., 2021)	CD	Pre,GTI	/	/	
SMOTE-LOF (Maulidevi et al., 2021)	F	Post	/		

Nevertheless, there is still a large area of aspect that can be observed and analyzed from studies related to SMOTE. Therefore, this studies attempts to investigate the relationship between data complexity and SMOTE variants with the objective to observe how the latter's performance will affect the changes in the former's measurement. Previous works have never associated data complexity in imbalanced datasets with SMOTE variants since they focus on IR aspect which is typically deemed as the prime issue related to imbalanced classification problem.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

This chapter is going to elaborate on the steps conducted for carrying out all three objectives as stated in Chapter 1. The proposed research design is presented in Section 3.2 where this section is separated into subsections that explain the steps required in the proposed imbalanced classification method. Lastly, this chapter is summarized in Section 3.3.

Figure 3.1 is illustrated to give an insight on the direction of this thesis especially on the designated methodology for every proposed research question. This figure also clearly point out the contributions of this study in proposing new SMOTE variant algorithm that includes complexity measure besides investigating the effect of complexity change in classification task.

3.2 Experimental Design

This study employs a quantitative approach since it requires experimental research to statistically analyze the classification performance of an oversampling technique. Besides, the type of data for prediction of the class label used is numerical data(Apuke, 2017).

There are several processes involved in the proposed experimental design for handling imbalance datasets as shown in Figure 3.2. Those processes are train-test split, oversampling, complexity measure, learning and performance measure. Details for each of the processes will be explained in the next subsection.

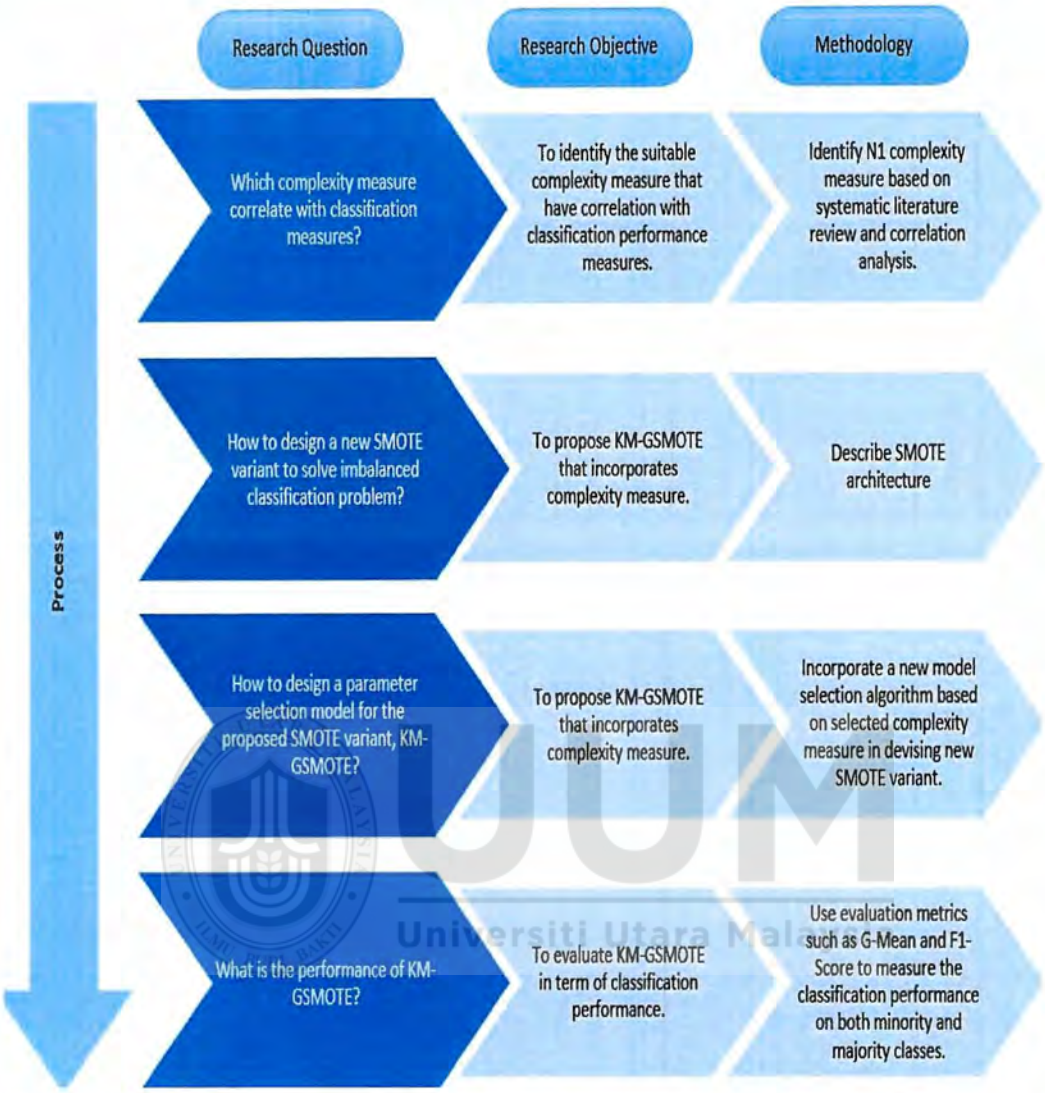


Figure 3.1. Specific methodology has been determined to resolve each of the research questions as stated in Chapter 1.

3.2.1 Train-Test Split

Train-test split is the initial step in any imbalanced classification and this phase is necessary to avoid any bias during evaluation of prediction performance. The machine learning model or classifier needs to have the ability to predict out-of-sample data, therefore the dataset must be splitted into training and test sets where the former is used to to train the model while the latter is for evaluation process. Test set is essential to assess whether the model can predict the unseen data that are not found during training and to measure the generalization ability.

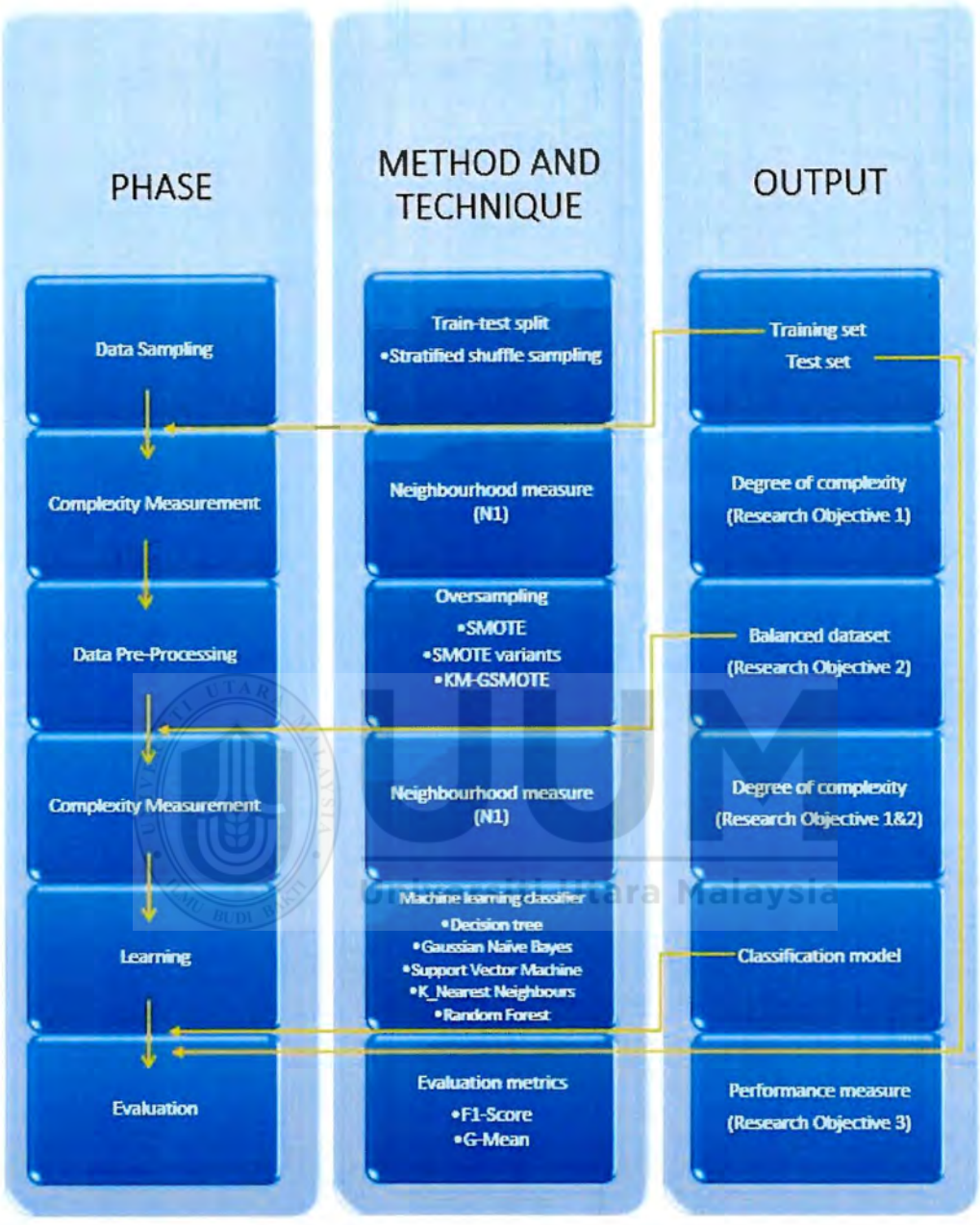


Figure 3.2. Research framework proposed in this research, divided into three categories, phase, method and technique, and output.

The ratio between training and test sets is set to be 7:3 which is a common practice in previously published researches. Studies conducted by (Van Dao et al., 2020; Nguyen et al., 2020, 2021) prove that 7:3 ratio is the best approach to train and validate machine learning model as well as to obtain the best performance. Since this study is focusing on the binary imbalanced class dataset, stratified shuffle sampling will be implemented

to make sure the ratio of examples from both majority and minority class are proportionally selected in every fold according to the imbalance ratio. Through stratification, the outnumbered and under-representative minority examples will be included in each fold without favoring majority class that has abundant examples.

3.2.2 Oversampling

Oversampling process adds synthetic minority examples into data space because minority class tends to suffer from under-representation and lack of information. In order to avoid the machine learning algorithm from being biased towards majority class and ignoring minority examples during learning process, the class distribution needs to be balanced out.

In this study, a new SMOTE variant that incorporate complexity measures is proposed. The original SMOTE algorithm will be altered accordingly such that the oversampling strategy attempts to reduce the degree of data complexity apart from improving the classification performance. Basically, synthetic examples are generated until the number of minority examples is the same as majority examples through linear interpolation between neighbouring minority examples, providing useful and additional information regarding the class.

Moreover, this variant will be capable of avoiding a significant increase in the data complexity. The new SMOTE variant is proposed considering that SMOTE is likely to worsen noisy environment as well as the class overlap and also does not able to tackle small disjuncts which eventually would intensify the degree of complexity. At the end of experiment, the results of all SMOTE variants as well as proposed variant will be compared to observe how well they improve the classification performance which reflected by the evaluation metric values and concurrently reduce the intrinsic

complexity despite formation of synthetic minority examples.

The parameters for every SMOTE variants used in the experimentation design are presented in Table 3.1.

Table 3.1

The parameters for SMOTE variants used in this thesis.

SMOTE Variant	**Parameters**
DBSMOTE (Bunkhumpornpat et al., 2012)	$p = 1.0$, $\varepsilon = 0.8$, $min\_samples = 3$
SMOTE-RSB* (Ramentol et al., 2012)	$p = 2.0$, $k = 5$
KM-SMOTE (Douzas et al., 2018)	$p = 1.0$, $k = 5$, $n\_clusters = 10$, $irt = 2.0$
KM-GSMOTE	$p = 1.0$, $k = 5$, $n\_clusters = 10$, $irt = 2.0$

The experiment uses default parameter values because iIt is usual practise to utilise default value parameters in open libraries since it enables users to rapidly begin experimenting with machine learning algorithms without having to define all of the configuration variables. The default values, determined by the library developers, are selected to offer sensible behaviour for the majority of use situations (Buitinck et al., 2013).

The selection of RSB*-SMOTE, DBSMOTE and KM-SMOTE are made on the basis that each of this SMOTE variants could tackle the data complexities like noisy environment, class overlapping and small disjuncts respectively. Furthermore, DBSMOTE and KM-SMOTE aggregate clustering technique with the oversampling algorithm similar to the proposed new SMOTE variant. DBSMOTE implements density-based spatial clustering of applications with noise (DBSCAN) where as KM-SMOTE uses k-means clustering.

3.2.3 Complexity Measure

In this experiment, the fraction of points on the class boundary (N1) will be calculated before and after implementation of oversampling strategy. This is done to observe to what extent SMOTE variants could affect the complexity of data besides understanding the performance of these variants.

The main reason behind selection of N1 complexity measure instead of others is because N1 could assess the difficulty of a dataset in terms of separation in distribution of examples from different classes. A measure that can reflect the complexity of decision boundary could be considered as robust due to the fact that it can discover the level of class overlapping as well as noisy environment which involves two dissimilar examples to be located near to each other that might cause confusion during classification task. By assessing the distribution of examples, specifically the ones that could bring about confusion to the learning algorithm, this study could generally comprehend the difficulty of dataset and eventually propose a better oversampling algorithm that would enhance the overall classification task.

3.2.4 Learning Algorithms

A classification model is trained on the transformed training set (balanced dataset) so that the learning algorithm (classifier) can discover the relationship between the input values and the class labels (output). The learning process is significant in order to proceed with classification task. This study adopts Gaussian Naive Bayes (GNB), Decision Tree (DT), Support Vector Machine (SVM), k-nearest neighbours (k-NN), and Random Forest (RF) classifiers to train the model and classify the class label as they have better classification ability compared to the others and commonly used in machine learning.

GNB and DT are both a supervised learning approach where the former is an extension of Naive Bayes which can be categorised as probabilistic machine learning algorithm while the latter generates models that look like trees. It use mathematics' if-then rule to generate sub-categories that fit into larger categories, allowing for exact, organic classification.

SVM is based on maximal-margin approach in which it maximizes the the margin between two different classes in binary classification so that the model could generalise better when predicting the unseen data. They operate by locating hyperplanes, which might be observed as a line dividing two distinct classes of data, inside a data distribution. The algorithm will choose the best line of separation from among the several hyperplanes that may typically be used to separate the data. The optimal hyperplane in the SVM model is the line that provides the widest margin between the various classes. SVM is great machine learning classifier because it get increasingly accurate as data input complexity increases.

On the other hand, *k*-NN is an instance-based model or also known as lazy-learning. It is a pattern recognition algorithm that retains training data points and learns from them by figuring out how they relate to other data in an n-dimensional space. The goal of *k*-NN is to locate the *k* closest connected data points in upcoming, unseen data. For the experimentation purpose, the number of cluster has been set as 1,3 and 5.

Lastly, a random forest is made up of several independent decision trees that work together as an ensemble. It attempts to produce an uncorrelated forest of trees whose forecast by committee is more accurate than that of any individual tree by using bagging and feature randomness while generating each individual tree. The class with the most votes becomes the class predicted by model, which is generated by the in-

dividual trees in the random forest. Table 3.2 shows the default parameters set for experimentation purpose.

Table 3.2

Parameters for classifiers used in this study.

Classifier	Parameters
GNB	$var\_smoothing = 10^{-9}$
k-NN	$k = \{1,3,5\}$
DT RF	$criterion = gini$, $min\_samples\_split = 2$
SVM	$C = \{10^{-3}, 10^{-2}, \ldots, 10^{3}, 10^{4}\}$, $\gamma = \frac{1}{\#\text{Features} \times Var(X)}$

3.2.5 Performance Measure

The classification model is then used on the test set to predict the class labels where the predicted labels are compared to the expected labels. The performance of this model is then evaluated by using performance measures.

As the discussion is revolving around imbalanced dataset, hence, it is highly appropriate to use a performance measure that can emphasize the importance of positive class during experimentation stage. Hence, G-Mean value and F1-Score are used as our performance measures.

Geometric mean (G-Mean) is a metric that measures the balance between classification performance on both the majority and minority classes. G-Mean has been extensively used in evaluating imbalanced classification problem in which it can be considered as replacement metric for accuracy. A low G-Mean indicates poor performance in minority examples classification, even if majority examples are accurately identified. This metric is critical for preventing overfitting the majority class and under-fitting the minority class.

Accuracy is not used in this study because a model could earn a high accuracy score since it accurately predicted the majority of dataset. However, this masks the model's real performance, which is objectively poor because it the prediction is done on one class only which is majority class that has abundance examples, ignoring the prediction of majority examples.

F1-Score, however is the harmonic mean of other measures called precision and recall. These performance measures are calculated using the numbers of correctly classified majority or minority class examples as well as the number of majority class examples misclassified as minority and vice versa. The definition of F1-score can be observed in Equation ??.

There are several measures for assessing classification systems. In this study, a specific set of criteria is used for evaluation since different metrics give varied and significant information into how each model operates. In this study, the metrics are used to assess and compare the classification performance of SMOTE variants and KM-GSMOTE in which any variant that yield higher G-Mean and F1-score is able to predict the minority and majority classes correctly by minimizing the false prediction.

3.3 Benchmark Datasets

For experimental design, the performance of machine learning classifiers will be evaluated based on 6 binary imbalanced datasets. Since multi-class imbalanced classification is out of the scope of this paper, only binary imbalanced datasets are being considered for experiments. Most of the datasets are originally multi-class imbalanced datasets obtained from the UCI Machine Learning Repository but they have been reorganized as binary imbalanced datasets to fit the purpose of this study. This is done by viewing the union of all other classes as the majority class, while choosing a limited

number of classes with low cardinality to represent the minority class.

Other researchers (Kunakorntum et al., 2020; Yuan et al., 2020; Kovács, 2019a) commonly use these datasets as the benchmark datasets when it comes to imbalanced classification problem. This is because of the variety of IR values, sizes of datasets, and even the amounts of minority examples in the datasets could portray the general performance of oversampling technique on datasets with diverse properties. Furthermore, the binary imbalanced datasets represent various applications such as healthcare (ecoli, arrhythmia), marine life (abalone), industry (wine_quality), environment (ozone_level) and so on. The datasets are tabulated in Table 3.3, and are sorted based on IR value in ascending order.

Table 3.3

Datasets used in the experiment, sorted by IR value in ascending order.

Dataset	#Instances	#Features	#Positive	#Negative	IR	Fischer's ratio	N1
ecoli	336	7	35	301	8.60	0.93087	0.11915
abalone	4177	10	391	3786	9.68	0.99586	0.02121
yeast_ml8	2417	103	178	2239	12.58	0.99860	0.19752
arrhythmia	452	278	25	427	17.08	0.99651	0.16772
wine_quality	4898	300	183	4715	25.77	0.99431	0.08197
ozone_level	2536	72	73	2463	33.74	0.98886	0.06535

The descriptions of each binary imbalanced dataset are as follows:

1. Ecoli dataset contains information about protein localization sites that is targeted as the predicted attribute. There are 8 classes of localization site. This study chooses one class to be the minority class which is inner membrane, uncleavable signal sequence (imU) class while the other classes are combined to be one majority class.
2. Abalone dataset is mainly used to predict age of abalone from physical measurements. The age of abalone is divided into 29 classes where class 7 has been

picked out as the minority class.

3. Arrhythmia dataset aims to identify the difference between the presence and absence of cardiac arrhythmia. The minority class is sinus bradycardia condition.
4. Red and white wine samples form the wine quality dataset where the quality depends on the physicochemical tests. The output wine quality is categorized based on the score in which data points with score equal and lower than 4 are labelled as minority class.
5. Ozone_level dataset provides information about wind, temperature and upwind ozone background level to forecast ozone and normal days where the former is the rare case and the latter is the typical case respectively.

3.4 Tools and Computer's Configuration

The experiment is conducted using python open libraries such as imblearn.datasets, sklearn.model_selection, gsmote, imblearn.over_sampling, sklearn and many more.

The computer's configuration is as follows:

1. Processor : Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz 2.11 GHz
2. Installed RAM : 16.0 GB
3. System type : 64-bit operating system, x64-based processor

3.5 Summary

The overall planning in the whole research design must be followed to achieve all research objectives as each of the processes influences one another. Every dataset is experimented using all SMOTE variants to observe the classification performance as well as the complexity measures.

The input for train-test split is the binary imbalanced datasets that are obtained from data repositories. The expected outputs from this process are training and test sets.

Then, in oversampling process, KM-GSMOTE is proposed such that the oversampling strategy incorporates data complexity to achieve a better performance than the original algorithm. After generation of synthetic minority examples in the data space which will balance the number of samples between majority and minority classes, the changes in complexity will be observed in order to assess the effect of oversampling on data complexity.

Next, complexity measure process provides an insight on the data intrinsic characteristics before and after oversampling. During learning process, the machine learning algorithm (classifier) will learn and predict the label of data points. Performance measure process will evaluate the classification performance of machine learning algorithm besides observing the predictive ability of the algorithm on unseen data.

From this evaluation, the performance of new SMOTE variant can be observed either it is better than SMOTE or not. The information gathered from complexity measures will be analyzed together with the result from performance measures in order to identify the correlation between these two as well as the impact of change in complexity has on the classification performance.

CHAPTER FOUR
KMEANS-GSMOTE

4.1 Introduction

This chapter discusses deeply on the proposed SMOTE variant method for imbalanced datasets. Section 4.2 will elaborate on the machine learning algorithm that motivated the proposal KM-GSMOTE algorithm which are *k*-means clustering and Geometric-SMOTE (GSMOTE). The steps involved in the proposed KMeans-GSMOTE (KM-SMOTE) algorithm are explained thoroughly in Section 4.2.

4.2 Motivation

4.2.1 *k*-Means Clustering

In data science, it is crucial for the researches to be able to reveal the underlying structure in the data by identifying the interesting patterns and distributions. This task can be performed using the clustering algorithm, which analyses data points that have similar characteristics to each other and groups them into a cluster without prior knowledge. The data points in a cluster should be different from the data points in the other cluster. The similarity index such as Euclidean, Manhattan, Minkowski, and Hamming distances can be used to measure the similarity between data points.

Clustering techniques are classified as unsupervised learning (Sinaga and Yang, 2020) as they are implemented on datasets that have no prior knowledge regarding the relationship between the unlabeled data points and they identify the patterns in the dataset instead of classifying nor labelling the data points which is usually done in supervised learning. Partitioning, hierarchical, density-based, distribution-based, and fuzzy clustering are the main clustering techniques in machine learning as shown in Figure 4.1.

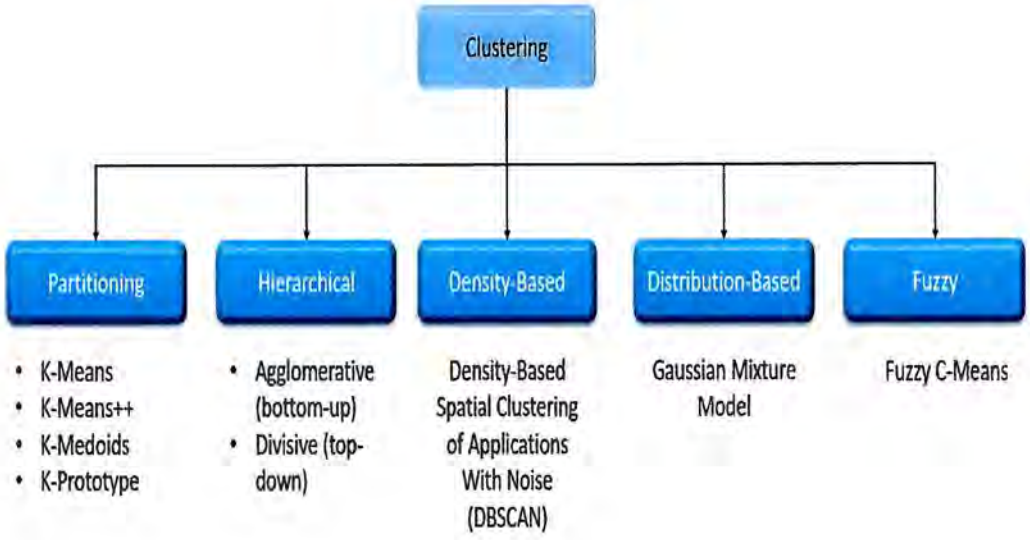


Figure 4.1. Categories of clustering techniques.

Partitioning clustering is exceptionally well-known apart from hierarchical clustering and k-means clustering is one of the most popular partitioning clustering techniques due to its simplicity, computation efficiency and usability (Zhao et al., 2018). k-means clustering has been widely used in real-world applications like banking to detect fraudulent transactions (Darwish, 2020), customer segmentation to identify buying behaviour and potential customers (Kansal et al., 2018), healthcare to categorize Alzheimer's disease patients (Alashwal et al., 2019), environmental issue to study spatial and temporal patterns of air pollutants (Govender and Sivakumar, 2020) and many others. Apart from that, k-means clustering's key benefits are it is adaptable to big data, appropriate for datasets with high feature dimensions and it also has a low reliance on the data (Wu et al., 2021).

Let $\mathbf{A} = \{a_1, \ldots, a_n\}$ be a dataset in a d-dimensional Euclidean space $\mathbb{R}^d$. Let $\mathbf{B} = \{b_1, \ldots, b_k\}$ be the k cluster centroids. Let $\mathbf{s} = [s_{im}]_{n \times k}$, where $s_{im} \in 0, 1$ showing if the data point a_i belongs to the $m-th$ cluster, $m = 1, \ldots, c$. The algorithm works by setting an initial set $\mathbf{B}$ of k cluster centroids and iteratively assigning the data points to their closest cluster centroid with the aim to minimize the sum of squared distances between

the cluster centroid and the data points within the cluster as shown in Equation 4.1.

$$J(s,B) = \sum_{i=1}^{n} \sum_{m=1}^{k} s_{im} ||a_i - b_m||^2 \tag{4.1}$$

The position of the centroids are updated until there is no significant change in the assignment of data point by using Equation 4.2 and Equation 4.3 where $||a_i - b_m||$ is the Euclidean distance between the data point a_i and the cluster centroid b_m. Eventually, the cluster centroid will be placed at the center of the respective cluster.

$$b_m = \frac{\sum_{i=1}^{n} s_{im} a_i}{\sum_{i=1}^{n} s_{im}} \tag{4.2}$$

$$s_{im} = \begin{cases} 1, & \text{if } ||a_i - b_m||^2 = \min\limits_{1 \le m \le k} ||a_i - b_m||^2 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

4.2.2 Geometric SMOTE

Geometric SMOTE (G-SMOTE) was developed as an improvement to the SMOTE data generation mechanism. The motivation behind this algorithm is to overcome some of the SMOTE algorithm shortfalls as follows:

- **The selection of k-nearest neighbours results in the formation of noisy examples**

  The value of k is predetermined in SMOTE, and the results of oversampling are sensitive to it in some cases, as shown in Figure 4.2. The decision boundary in Figure 4.2 denotes areas of the input space where instances from either the positive or negative class predominate. A few minority class instances, known

as noisy examples, that are near the decision boundary infiltrate the majority class area. Since x', the selected k-nearest neighbour of x, could be one of the previously mentioned noisy examples, a large k value might cause the generation of additional noisy examples.

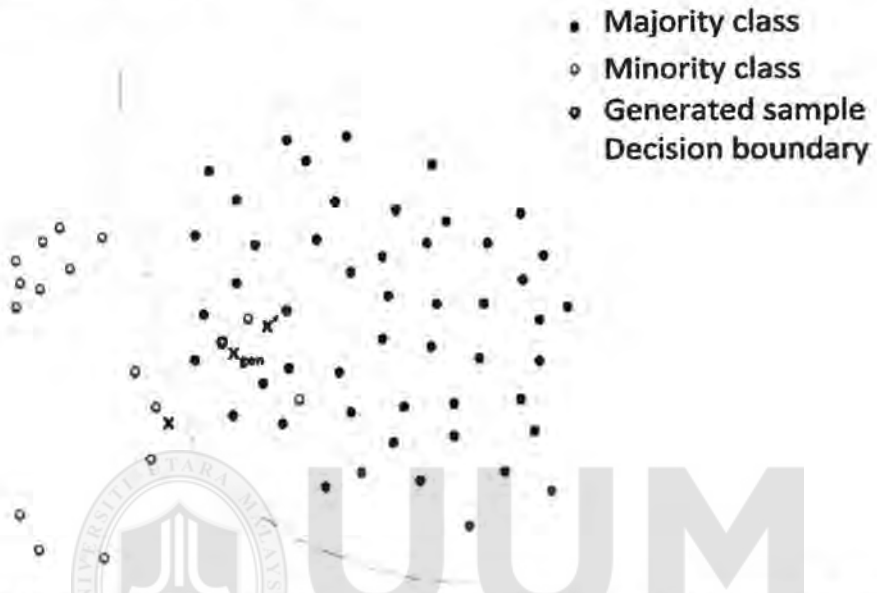


Figure 4.2. A noise example is generated by linear interpolation between an example located near the decision boundary and one of its randomly selected 4-nearest neighbours (Douzas and Bacao, 2019).

- **SMOTE initialisation results in formation of noisy examples**

  Setting a low k-nearest neighbour value does not prevent the probability of generating noisy examples when x itself in the first place is a noisy example as shown in Figure 4.3.

- **Formation of almost identical examples**

  Minority and majority regions can be grouped together in several clusters, each with its own set of decision boundaries. In this example, choosing k carefully may be a good technique because a small value can reduce the likelihood of generating noisy examples. On the other hand, as exemplified in Fig. 4.4 , where x and x' are located in the same cluster, it may increase the likelihood of generating synthetic examples in concentrated minority class areas. These duplicated synthetic examples are less effective because they do not offer new

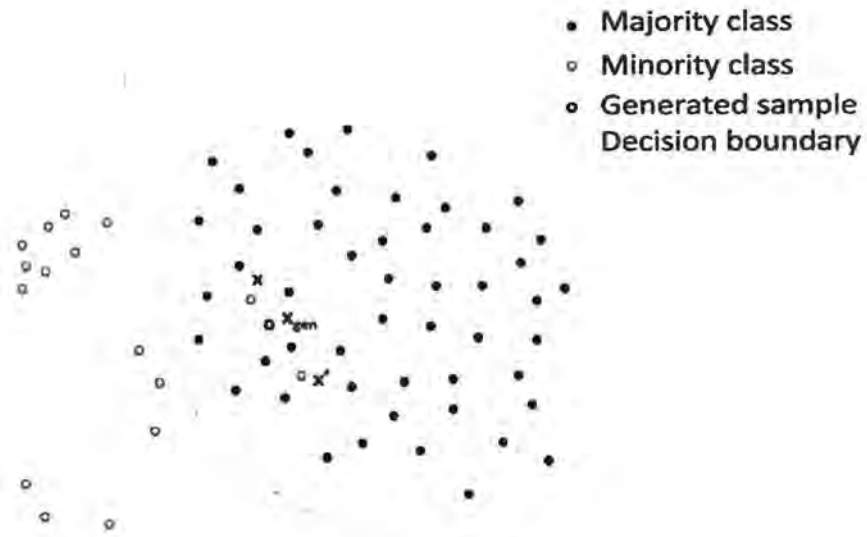


Figure 4.3. Even if the *k*-nearest neighbour value is set to 2, it cannot eliminate the formation of another noise example when a noise example is randomly selected as the initial observation (Douzas and Bacao, 2019).

information to the data set and promote over-fitting. As a result, expanding the data generation process in areas where minority examples are lacking is ideal.

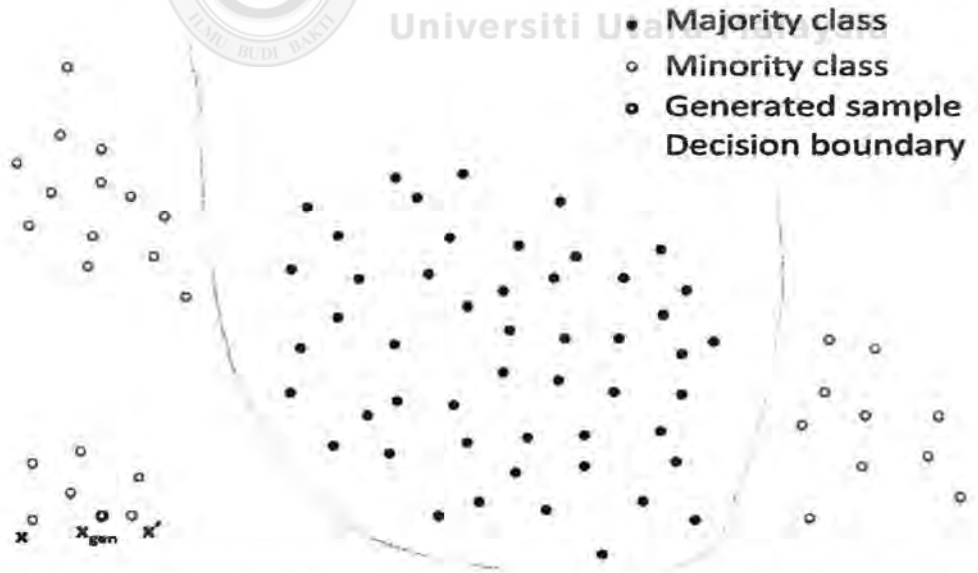


Figure 4.4. A duplicated synthetic example is generated when one of the 5-nearest neighbours of the selected minority example is located in the same cluster (Douzas and Bacao, 2019).

- **Initial observations from two different minority clusters might cause the generation of noisy examples**

To prevent the previous scenario, increasing k could very well result in a chosen x' such that x and x' came from different clusters. As shown in Fig. 4.5, this situation can probably generate new example inside the majority region.

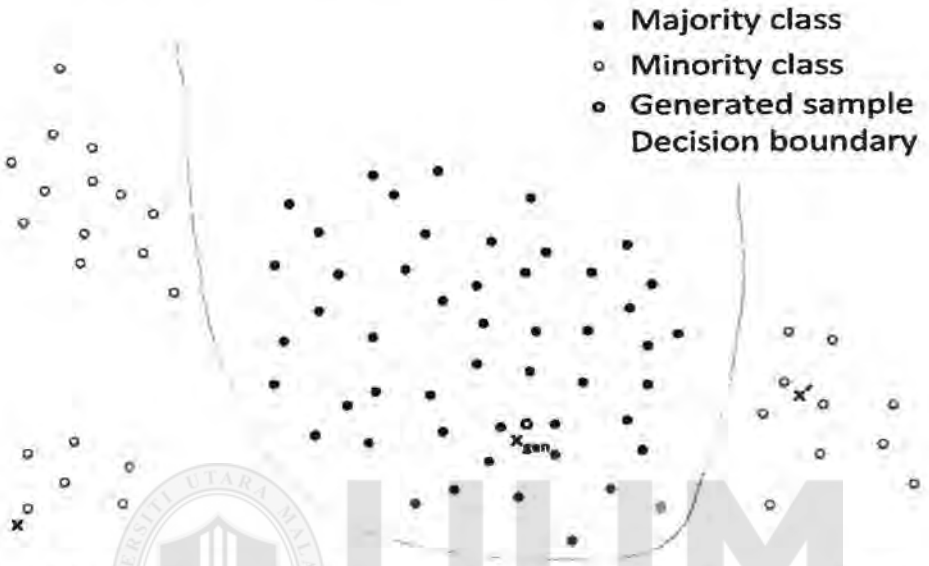


Figure 4.5. A noise example is generated within the majority region despite of attempting to generate an inter-cluster example by defining a large value of k-nearest neighbours (Douzas and Bacao, 2019).

G-SMOTE generates synthetic samples in a geometric region of the input space centred on each chosen minority example. While this region is a hyper-sphere in its default configuration, G-SMOTE allows it to be deformed to a hyper-spheroid. Since the shape of the deformed geometric region changes depending on the set of hyper-parameters, it allows the generation of synthetic of examples to be greatly diversified compared to SMOTE (Fonseca et al., 2021).

Apart from that, G-SMOTE improves the neighbour selection strategy of the SMOTE algorithm by not limiting it to based only on the minority class which occurs in SMOTE, but G-SMOTE also considers majority and combined neighbour selection strategy. The minority class region can be expanded aggressively without concerning the generation of noisy examples in majority class region in which the expansion is

constrained by the presence of closest majority or minority example that initially selected as a surface point, ensuring the examples are located within safe region (Douzas et al., 2021). Eventually, G-SMOTE was empirically proved to considerably improve the SMOTE oversampling performance (Douzas and Bacao, 2019).

4.3 KMeans-GSMOTE

By leveraging the benefits of both k-means clustering and G-SMOTE, a new SMOTE variant KMeans-GSMOTE (KM-GSMOTE) is proposed to improve the oversampling performance. KM-GSMOTE comprises of five steps as illustrated in Figure 4.6 which are measuring, clustering, filtering, oversampling, and decision-making.

Data complexity is calculated in measuring step to observe the level of intrinsic complexity of dataset. The initial complexity value observed during this step will be compared with the value after oversampling in the last decision-making step. This study computes N1 value that is obtained by constructing a class-blind minimum spanning tree (MST) that links all examples in a dataset based on pairwise distances. Following that, the number of examples that are connected to example from opposite class is computed. These occurrences are probably on the borderline or in overlapping areas and the final N1 measure is the ratio of their number to the entire number of instances. N1 is set between 0 and 1, with value closer to 0 indicates a simplification in defining the class boundary.

During clustering step, k-means clustering is used to divide the input space into k clusters. The k value is decided by the user, thus in this work k is set as $k = \{2,4,6,8\}$. The initial selection of k cluster centroids is randomly decided. Then, the examples in the dataset are assigned to their nearest cluster centroids. Then, the location of the centroids are updated in order to place these centroids in the center of the examples

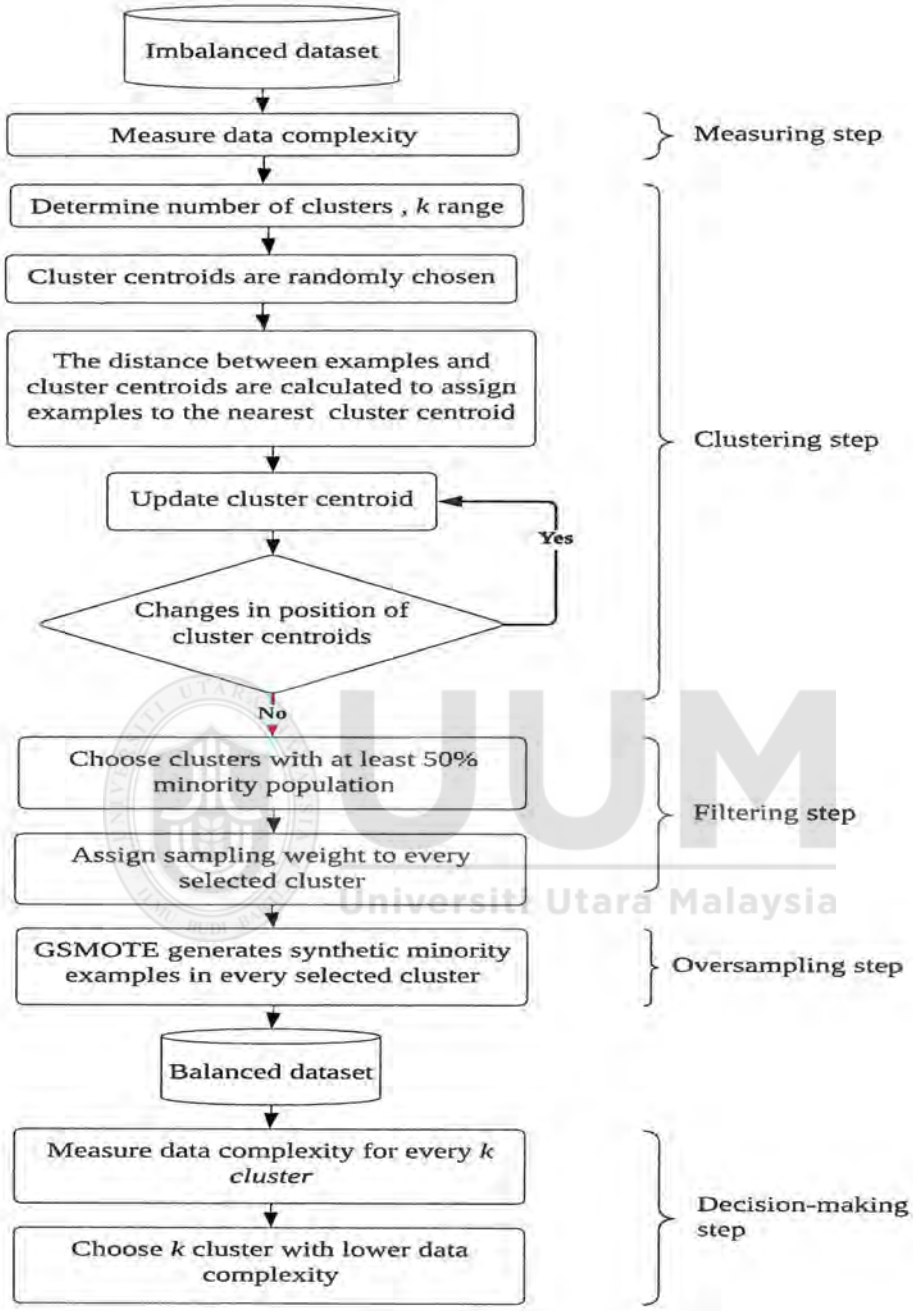


Figure 4.6. KM-GSMOTE algorithm flowchart

that form a cluster beforehand. The assignment of examples and the update of cluster centroids are repeated until no more examples can be reassigned which implies the algorithm is already converged.

The filtering stage, which comes after clustering, selects which clusters will be oversampled and decides how many examples should be generated for each cluster. The purpose of this step is to oversample just clusters that are dominated by the minority class because implementing GSMOTE in these areas is safer and less likely to cause noise generation. Besides, the objective is to acquire a balanced distribution of samples within the minority class. Therefore, the filter step gives sparse minority clusters more generated examples than dense ones thereby overcoming the within-class imbalance problem that correlates with small disjuncts.

The proportion of minority and majority examples in each cluster is used in choosing the desired cluster. Any cluster with minority examples constitute at least half of the population is automatically chosen for oversampling. The imbalance ratio threshold (or irt) is a hyperparameter of KM-GSMOTE with a default value of 1, can be changed to alter this behaviour. A cluster c imbalance ratio is expressed as:

$$\text{imbalance ratio(c)} = \frac{\text{majority(c)} + 1}{\text{minority(c)} + 1} \tag{4.4}$$

Cluster selection becomes more selective and demands a higher percentage of minority examples when the imbalance ratio threshold is raised. The selection criteria, on the other hand, become more flexible when the threshold is lowered, making it possible to select clusters with a bigger majority share.

Filtered clusters are given sampling weights between zero and one to define the distribution of samples to be generated. A high sampling weight produces more generated samples and is correlated with a low density of minority samples. This is accomplished by basing the sample weight on the ratio of the density of a single cluster to the av-

erage density of all picked clusters. It should be noted that just the distances between minority occurrences are taken into account when calculating a cluster's density.

Sampling weight with range [0,1] is allocated for every chosen clusters. The value of sampling weight depends on the density and sparsity of minority examples in a cluster. The formulas of these two measurement are shown in Equation 4.5 and Equation 4.6 where f indicates the number of features. More synthetic examples will be generated in less dense cluster compared to cluster with high density of minority examples. The sampling weight can be computed using five sub-computations:

1. Euclidean distance matrix is calculated for every selected cluster, f by excluding majority samples.
2. Adding up the distance matrix's non-diagonal elements, then dividing by the total number of non-diagonal elements, will get the mean distance inside each cluster.
3. To calculate the density measure of every cluster, the total of minority examples is divided by the average minority distance raised to the power of the number of features:

$$\text{density} = \frac{\text{number of minority examples}}{\text{average distance between minority examples}^{f}} \tag{4.5}$$

4. To obtain a measure of sparsity, the density measurement is inverted :

$$\text{sparsity} = \frac{1}{\text{density}} \tag{4.6}$$

5. Each cluster's sample weight is calculated by dividing its sparsity factor by the total sparsity factor for all clusters.

Consequently, the total sampling weights equal one. This feature allows the number of examples to be generated in each cluster to be calculated by multiplying the sampling weight of each cluster by the total number of examples to be generated.

In the oversampling step, GSMOTE is implemented in every selected clusters to introduce synthetic minority examples in a geometric region of the input space. At this point, the proportion between majority and minority classes is balanced which indicates the imbalanced problem has been resolved. Each cluster's points are passed to the oversampling algorithm, together with the directive to create $\|samplingweight(f) \times n\|$ samples, where n is the total number of samples to be produced. GSMOTE parameters constitute of truncation factor, deformation factor, selection strategy and number of nearest neighbors.

The G-SMOTE algorithm as shown in Algorithm 2 can be explained as follows:

1. Create an empty set S_{gen}.
2. The S_{min} components are rearranged, and the procedure listed below is carried out N times, up to the point when N synthetic examples have been produced.
3. The centre of a geometric area is chosen to be an example of a minority class called x_{center}. After rearranging the dataset in the previous phase, the order of the selection matches the order of the dataset examples. Therefore, some of the minority class examples will be chosen more than once if N is bigger than S_{min}.
4. At first appearance, this step appears to generalise the SMOTE algorithm's selection phase. More specifically, it leads to an example called $x_{surface}$ that is chosen at random and may or may not belong to the minority class, depending on the values of α_{sel}, k, and x_{center}. However, as it does more than only filter the chosen minority class examples based on heuristic guidelines like SMOTE

variants, it can be distinguished as a component of the G-SMOTE data generation method. On the basis of the neighbour selection strategy α_{sel}, three distinct cases might be identified as follows:

a) Case $\alpha_{sel} = minority$:

Only the minority class is used in the neighbour selection technique, which is the same as the SMOTE selection strategy. First, the set S_{min}'s k nearest neighbours of x_{center} are found, and one of them, $x_{surface}$, is chosen at random. A sample minority class instance selection among the k = 4 closest neighbours of the x_{center} is shown in Figure 4.7. This selection strategy's time complexity relies on the method used and rises as the input space's dimension and the value of the k parameter grow. Because of this, it is more computationally efficient to limit the search for nearest neighbours to members of the minority class rather than including examples of the majority class in the search space for a large number of realistic occurrences.

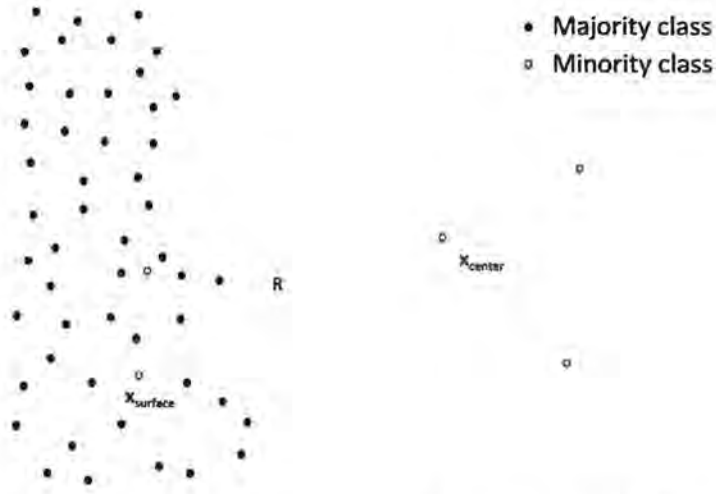


Figure 4.7. In minority selection strategy, the hyper-spheroid's centre is selected among minority examples, and one of its $k = 4$ nearest neighbours is chosen as the surface point. It is shown that the distance of these minority examples is equal to the hyper-spheroid's radius, R (Douzas and Bacao, 2019)

b) Case $\alpha_{sel} = majority$:

The minority selection strategy has the potential to produce synthetic minority example that penetrates the majority class region, which is one of its shortcomings. The majority selection method rules off this possibility by setting $x_{surface}$ from the set S_{maj} as the closest neighbour of x_{center}. Therefore, it is assured that when a random minority class point is formed inside a hypersphere with a centre x_{center} and a radius $R = ||x_{center} - x_{surface}||$, its distance from x_{center} is within the distance between x_{center} and any example from the majority class. However, since any information pertaining to the minority class is disregarded, this strategy may induce noisy environment due to vigorous expansion of the minority class region. Figure 4.8 shows the selection of closest majority example from x_{center}'s majority class neighbours. The downside of this strategy is the computational cost may be greater than that of minority selection strategy, particularly for datasets with high IR values.

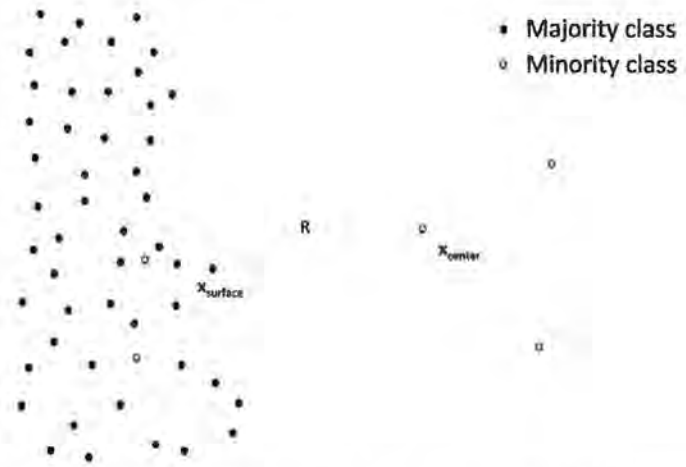


Figure 4.8. In majority selection strategy, the hyper-spheroid's centre is selected among minority examples, and the surface point is chosen from its nearest majority examples neighbour. It is shown that the distance of these examples is equal to the hyper-spheroid's radius, R (Douzas and Bacao, 2019)

c) Case $\alpha_{sel} = combined$:

The minority and majority selection techniques are first applied via the combined selection strategy, which then identifies x_{min} and x_{maj} as the chosen minority and majority class instances, respectively. In order to minimise its distance from the centre, x_{center}, the surface point $x_{surface}$ is determined to be either x_{min} or x_{maj}. Both of these circumstances, in which $x_{surface}$ is classified as either a minority class instance or a majority class instance, are shown in Figures 4.9 and 4.10. Following the combined selection method, the closest majority class neighbour of the centre limits the extension of the minority class region relative to the selected as a centre minority class sample, preventing the formation of noisy samples. The expansion is not only secured, but it is additionally constrained by the existence of minority class instances, in contrast to pure majority selection technique. The disadvantage of the combined is that it has a larger computational cost than the SMOTE/minority selection approach. This is analogous to the disadvantage of the majority selection technique.

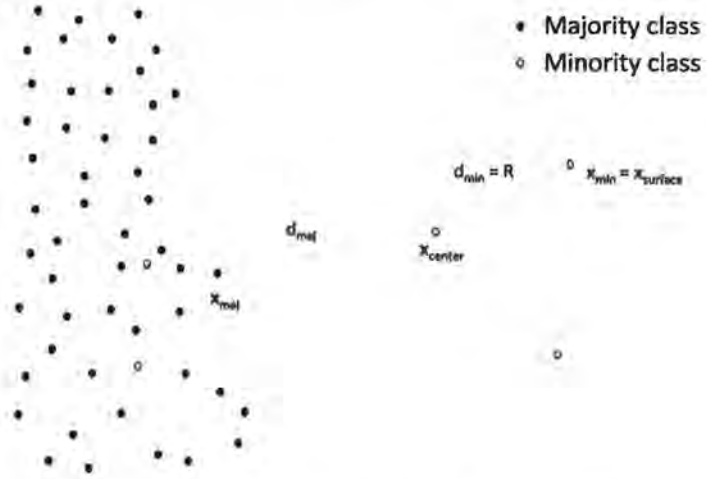


Figure 4.9. The nearest example, either from minority or majority class is determined to be the surface point. In this case, a minority examples has been selected. (Douzas and Bacao, 2019)

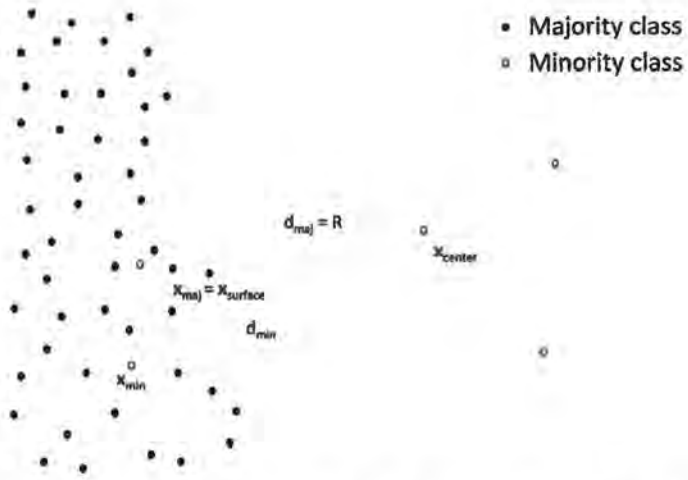


Figure 4.10. Since it is closer to the centre than the chosen example from the k closest minority class neighbours of the centre, the majority class sample that is closest to the centre is designated as the surface point. (Douzas and Bacao, 2019)

5. There are two unique directions formed in the input space: $x_{//}$ and $x_{\perp}$. The former shows how x_{gen} is projected onto the unit vector $e_{//}$ of equation from line 41 which joins x_{center} and $x_{surface}$, whereas the latter is perpendicular to that vector and likewise belongs to the hyperplane created by x_{gen} and $e_{//}$.
6. The generation of synthetic minority examples starts from this step. By applying Equation 4.7 to a unit hyper-sphere centred at the input space's origin, a random point e_{sphere} is produced on its surface. Equation in line 38 is used to convert the point e_{sphere} into a randomly generated example x_{gen} inside the unit hyper-sphere. Figure 4.11 illustrates randomly generated example that is evenly dispersed inside the unit hyper-sphere is the process's end outcome.

$$e_{sphere} = \frac{v_{normal}}{||v_{normal}||} \tag{4.7}$$

$v_{normal} \leftarrow (v_1, ..., v_p)$ is a vector generated from p random numbers from the normal distribution $N(0,1)$.

7. The generated example x_{gen} is subjected to a transformation in this stage. As was stated before, the unit vector $e_{//}$ of equation in line 31 reflects the special direc-

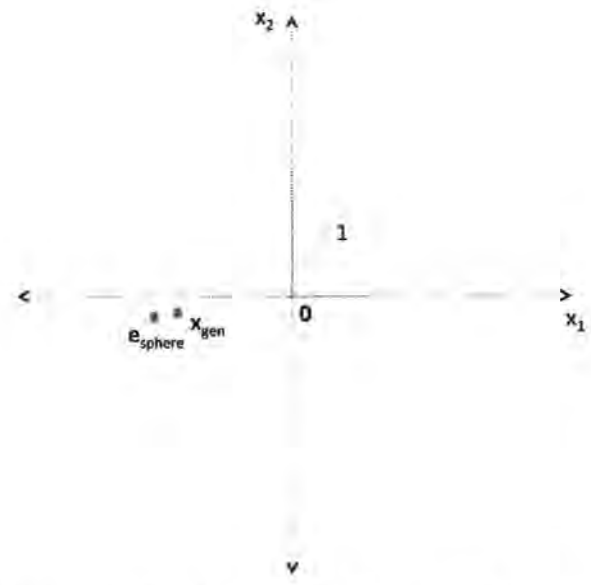


Figure 4.11. The input space's origin acts as the center for a unit hyper-sphere. On the surface, a point is generated at random and moved inside the unit hypersphere (Douzas and Bacao, 2019).

tion that is formed in the input space by the centre x_{center} and the chosen surface point $x_{surface}$. By creating synthetic examples along the line segment between x_{center} and $x_{surface}$, the SMOTE mechanism, which always chooses a minority example as a surface point, takes advantage of this direction. A generalised version of the SMOTE mechanism is modelled by the G-SMOTE algorithm. More particularly, a family of parallel hyper-planes that are perpendicular to the unit vector $e_{//}$ is defined. A linear mapping between α_{trunc} and the point where each hyperplane and the parallel to $e_{//}$ diameter meet is constructed. As a result, each of these hyper-planes divides the interior of the hypersphere into two sections and correlates to a certain value of α_{trunc}. Allow P to be the hyper-plane that traverses the origin and P' to be the hyper-plane for a particular non-zero value of α_{trunc}. When $\alpha_{trunc} > 0$, the interior of the hypersphere is truncated to contain just the section that does not include the $e_{//}$ point; if the x_{gen} point is a part of that section, it is mapped to the symmetric point of equation in line 42, where $x_{//}$ is specified. If the x_{gen} is in the truncated region, condition in line

41 is checked. A sample of the transformation previously described is shown in Figure 4.12. In the scenario where $\alpha_{trunc} < 0$, the transformation is described identically, except in this instance the truncation takes place in the region that contains the $e_{//}$ point. The degree of truncation is determined in both situations by the hyper-parameter α_{trunc}'s absolute value. The truncated hyper-sphere regions for various positive and negative values of α_{trunc} are shown in Figure 4.13. The above transformation significantly modifies the original uniform probability distribution in the hyper-sphere, which is the last thing to note. The truncated area's p.d.f. value is zero, but that of its P-symmetric mapped area is doubled.

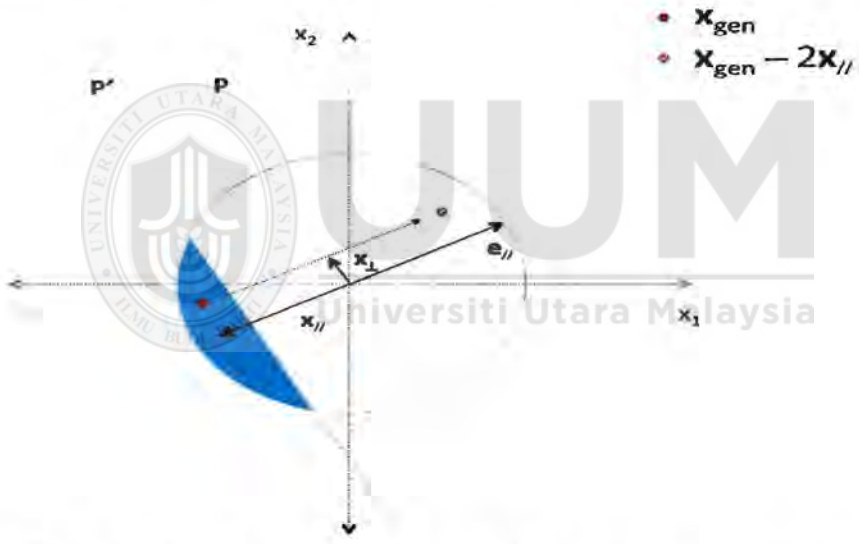


Figure 4.12. This figure shows the application of Truncate transformation where the resulting truncated region is shown by the shaded region (Douzas and Bacao, 2019).

8. This phase explains a transition that takes place when the hyper-sphere deforms into a hyper-spheroid. More specifically, the example x_{gen} is positioned in the direction of the parallel to $e_{//}$ diameter, perpendicular to the unit vector $e_{//}$. The α_{def} hyper-parameter regulates this mapping, and from equation in line 45 varies proportionally with it. Any point on the surface of the hyper-sphere will thus stay on that surface even after all axes, other than the one described by the $e_{//}$ unit vector, have been scaled by the factor α_{def}. The construction of a hyper-

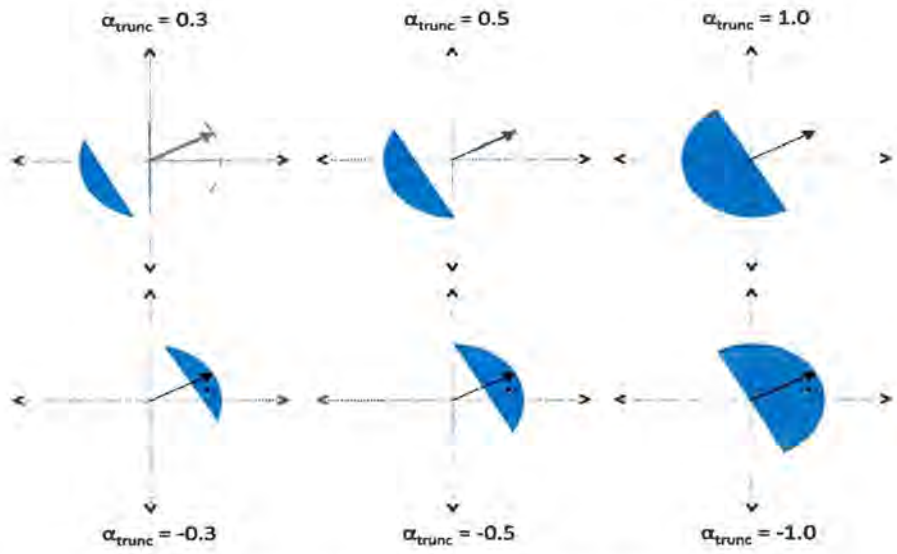


Figure 4.13. Regions that have been truncated for different values of α_{trunc} (Douzas and Bacao, 2019).

spheroid border with the symmetry axis in the $e_{//}$ direction essentially results from this. The original uniform probability distribution is further altered by the deformation transformation, much like the truncation. The mapping of the x_{gen} example as a result of the unit hyper-deformation sphere's is shown in Figure 4.14 . Increased α_{def} values have an impact on the hyper-sphere deformation, as seen in Figure 4.15.

9. The generated example is scaled by the radius R value and translated by the x_{center} vector in the algorithm's last step. Equation in line 47 provides a description of the combined outcome of these two transformations. Following the application of truncation, deformation, and translation, Figures 4.16 and 4.17 illustrate the resulting limits of the region where data can be generated as well as a randomly generated example x_{center} for the two distinct situations shown in the Figures 4.9 and 4.10. Algorithm 2 describes KM-GSMOTE.

Lastly, the complexity level of balanced dataset for every k value is calculated to decide which k cluster yields lower complexity measurement after oversampling besides

Algorithm 2 KM-GSMOTE

Input:

$\mathbf{D} \leftarrow \{d_1, d_2, \ldots, d_n\}$ ▷ Set of n data points

$k_{cluster} \leftarrow$ *Number of cluster*

$S_{maj} \leftarrow$ *Majority class samples*

$S_{min} \leftarrow$ *Minority class samples*

$N \leftarrow$ *Total number of synthetic examples to be generated*

$k \leftarrow$ *Number of nearest neighbors*

$\alpha_{sel} \leftarrow$ *The neighbour selection strategy where* $\alpha_{sel} \in$ {minority, majority, combined}

$\alpha_{trunc} \leftarrow The truncation factor where -1 \leq \alpha_{trunc} \leq 1$

$\alpha_{def} \leftarrow The deformation factor where 0 \leq \alpha_{def} \leq 1$

Output:

$S_{mincluster} \leftarrow$ *Set of selected clusters*

$S_{gen} \leftarrow$ *Set of generated synthetic examples*

$S_{mincluster} \leftarrow$ *Set of selected clusters*

procedure CLUSTER(D,$k_{cluster}$)
  for each iteration l **do**
    compute r_{nk}:
    for each data point in D **do**
      Assign each data point to a cluster:
      for each cluster, $k_{cluster}$ **do**
        if $k_{cluster} = argmin||d_n - \mu_k^{l-1}||$ **then**
          $r_{nk} = 1$
        else
          $r_{nk} = 0$
    for each cluster, $k_{cluster}$ **do**
      Update cluster centers as the mean of each cluster:
      $\mu_k^l = \frac{\sum r_{nk} d_n}{\sum r_{nk}}$
  $S_{mincluster} = x_{min} \in$ each cluster, $k_{cluster}$, where $\sum x_{min} \geq 50\%$ of the population.
  for $s \in S_{mincluster}$ **do**

    density (s) $= \frac{\sum x_{min}}{\text{average distance between} x_{min}^{'}}$

    sparsity (s) $= \frac{1}{density(s)}$

    weight (s)$= \frac{sparsity(s)}{\sum sparsity(S_{mincluster})}$

procedure SURFACE(α_{sel}, x_{center}, S_{maj}, S_{min})
  if $\alpha_{sel} = minority$ **then**
    $x_{surface} \in S_{min,k}$
  else if $\alpha_{sel} = majority$ **then**
    $x_{surface} \in S_{maj,1}$
  else if $\alpha_{sel} = combined$ **then**
    $x_{min} \in S_{min,k}$
    $x_{maj} \in S_{maj,1}$
    $x_{surface} \leftarrow \text{argmin}_{x_{min}, x_{maj}}(||x_{center} - x_{min}||, ||x_{center} - x_{maj}||)$
  return $x_{surface}$

procedure VECTORS(x_{center}, $x_{surface}$)
  $e_{//} \leftarrow \frac{x_{surface} - x_{center}}{||x_{surface} - x_{center}||}$
  $x_{//} \leftarrow (x_{gen} \cdot e_{//}) e_{//}$
  $x_{\perp} \leftarrow x_{gen} - x_{//}$
  return $x_{//}, x_{\perp}$

procedure HYPERBALL
  $v_i \sim N(0,1)$
  $r \sim U(0,1)$
  $x_{gen} \leftarrow r^{1/p} \frac{(v_1, \ldots, v_p)}{(||v_1, \ldots, v_p||)}$
  return x_{gen}

procedure TRUNCATE(α_{trunc}, x_{gen}, x_{center}, $x_{surface}$)
  if $|\alpha_{trunc} - x_{//} > 1|$ **then**
    $x_{gen} \leftarrow x_{gen} - 2x_{//}$
  return x_{gen}

procedure DEFORM(α_{dist}, x_{gen}, x_{center}, $x_{surface}$)
  return $x_{gen} - \alpha_{def} x_{\perp}$

procedure TRANSLATE(x_{gen}, x_{center}, R)
  return $x_{center} + R x_{gen}$

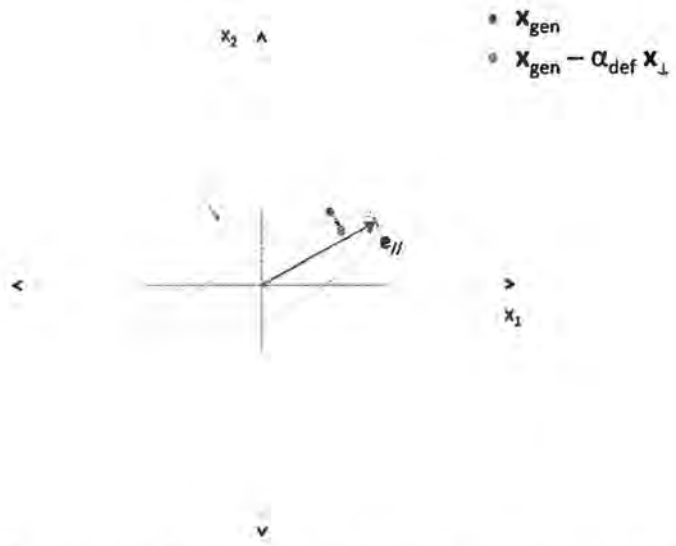


Figure 4.14. The generated point is transferred to a new point in the direction of the hyper-diameter sphere's after the transformation Deform is applied (Douzas and Bacao, 2019).

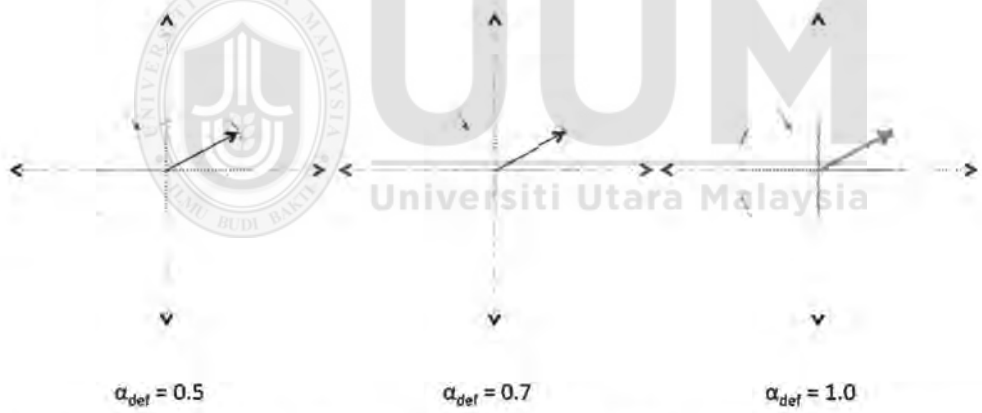


Figure 4.15. The impact of raising α_{def} on hyper-sphere deformation. The last instance is a deformation of a line segment hyper-sphere (Douzas and Bacao, 2019).

achieving a very good classification performance as seen in Algorithm 3. This can be done by doing comparison between N1 values after oversampling and before oversampling. Computation of complexity measure is necessary in decision-making step for this algorithm to be able to choose *k* cluster that could reduce the intrinsic complexity of the balanced dataset remarkably despite addition of synthetic examples.

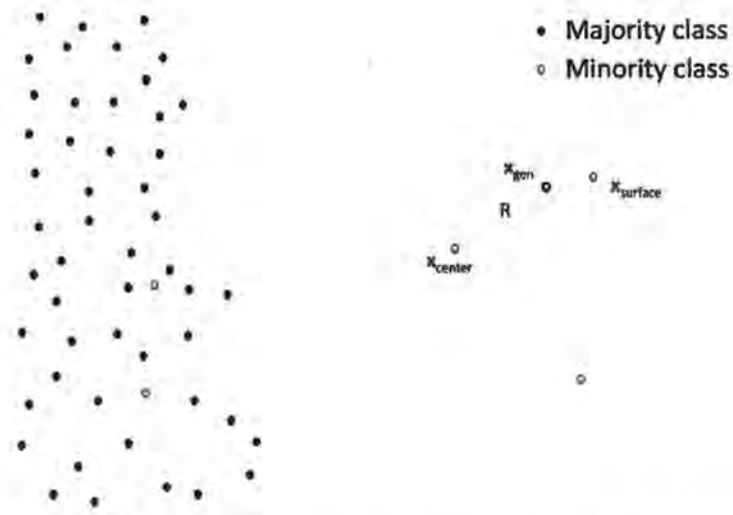


Figure 4.16. Limits of the region where data can be generated in the scenario shown in Figure 4.9 (Douzas and Bacao, 2019).

4.4 Summary

This chapter presented the proposed SMOTE variant, KM-GSMOTE which is a straightforward and efficient oversampling technique based on utilizing k-means clustering algorithm and G-SMOTE. The justification behind implementation of k-means clustering are this technique has seen extensive use in practical applications besides having a minimal dependency on the data, ideal for datasets with large feature dimen-

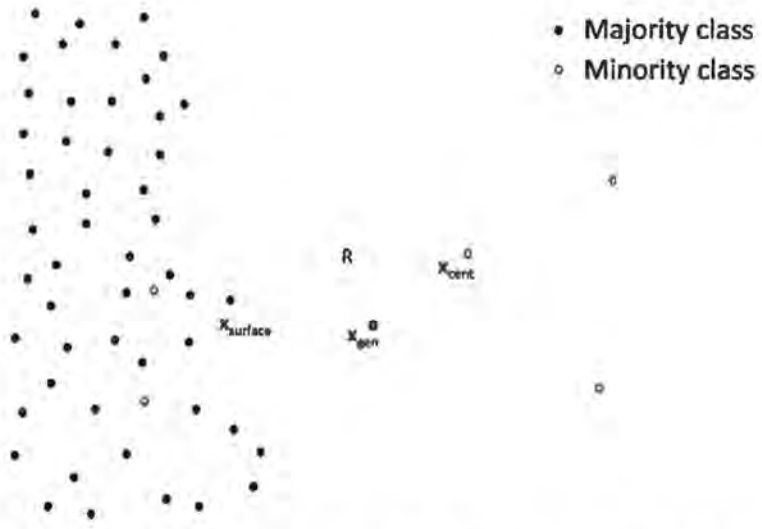


Figure 4.17. Limits of the region where data can be generated in the scenario shown in Figure 4.10 (Douzas and Bacao, 2019).

48: **for** $s \in S_{mincluster}$ **do**
49: $S_{gen} = \emptyset$
50: **while** $|S_{gen}| < N$ **do**
51: $\mathbf{x}_{center} \in S_{min}$
52: $\mathbf{x}_{surface} \leftarrow SURFACE(\alpha_{sel}, \mathbf{x}_{center}, S_{maj}, S_{min})$
53: $(\mathbf{x}_{//}, \mathbf{x}_{\perp}) \leftarrow VECTORS(\mathbf{x}_{center}, \mathbf{x}_{surface})$
54: $\mathbf{x}_{gen} \leftarrow HYPERBALL()$
55: $\mathbf{x}_{gen} \leftarrow TRUNCATE(\alpha_{trunc}, \mathbf{x}_{gen}, \mathbf{x}_{center}, \mathbf{x}_{surface})$
56: $\mathbf{x}_{gen} \leftarrow DEFORM(\alpha_{dist}, \mathbf{x}_{gen}, \mathbf{x}_{center}, \mathbf{x}_{surface})$
57: $\mathbf{x}_{gen} \leftarrow TRANSLATE(\mathbf{x}_{gen}, \mathbf{x}_{center}, ||\mathbf{x}_{center} - \mathbf{x}_{surface}||)$
58: $S_{gen} \leftarrow S_{gen} \cup \{\mathbf{x}_{gen}\}$

Algorithm 3 Model Selection for KM-GSMOTE

Input:
G ← *Undirected, connected graph* ▷ Connected examples regardless of the class
r ← *Root*
V ← *Vertices*

Output:
MST ← *Minimum spanning tree*

1: **for** $u \in V[G]$ **do**
2: $key[u] = \infty$
3: $\pi[u] = NIL$
4: key[r] = 0
5: Q = V(G)
6: **while** Q is not empty **do**
7: u = EXTRACT-MIN(Q)
8: **for** $v \in Adj[u]$ **do**
9: **if** $v \in Q$ and w(u,v) < key[v] **then**
10: DECREASE-KEY($Q, key[c], w(u,v)$)
11: $\pi[v] = u$
return $(v, \pi[v]) : v \in V - r$
$\mathrm{N1} = \frac{1}{n}\sum_{i=1}^{n} I((\mathbf{x}_i, \mathbf{x}_j) \in \mathrm{MST} \wedge y_i \neq y_j)$

sions, and can be adjusted to big data. G-SMOTE is generally an improved version of SMOTE that overcome SMOTE data generation mechanism's downsides. SMOTE does not alter the class-specific mean values on the high-dimensional data while reducing data variability and generating sample correlation. By establishing a geometric region where the data production process takes place, G-SMOTE, in contrast, broadens the scope of the linear interpolation mechanism. This geometric area of the input space is a truncated hyper-spheroid at the widest sense choice of hyper-parameters.

Next, this chapter describes clearly the steps involved in KM-GSMOTE algorithm namely measuring step, clustering step, filtering step, oversampling step, and decision-making step. The contribution of this study can be distinguished during measuring, oversampling and decision-making steps. Measuring step provides useful informa-

tion on the intrinsic data complexity level which later will be compared with the final data complexity level after implementation of oversampling in decision-making step. The acquired result would give an insight on the changes in complexity level of every model whereby the model with lower complexity value and better classification performance can be selected as the final output. The correlation between complexity measure and classification performance can be identified from the previous result.

CHAPTER FIVE
RESULT & ANALYSIS

5.1 Introduction

In this section, the analysis of obtained results is divided into two parts. The first part is for performance analysis based on SMOTE and classifiers. In this part, G-Mean and F1-Score are used to evaluate the capability of each classifier. Next, in the second part, the complexity analyses of each SMOTE variant are carried based on Fischer's ratio complexity and N1 complexity.

The experimental results will provide the answer for all research questions and objectives as stated in Chapter 1 regarding identification of complexity measures that are likely to correlate with classification measures, utilization of complexity measures to improve the classification performance of proposed SMOTE variant (KM-GSMOTE) and performance evaluation of KM-GSMOTE.

5.2 SMOTE Variant-Classifier Performance Analysis

This section will describe the performance of five classifiers tested over three existing SMOTE variants which are DBSMOTE, SMOTE-RSB* and KMSMOTE as well as the proposed variant, KM-GSMOTE. The performance is analyzed based on G-Mean and F1-Score taken from each experiment.

To show how balance the classification performance is, for each classifier, the difference between the G-Mean of each SMOTE variant for each dataset with the G-Mean without SMOTE (baseline) is calculated, and then divided by the G-Mean of the baseline, as in Equation 5.1.

$$\Delta\text{G-Mean} = \frac{\text{G-Mean in SMOTE Variant} - \text{G-Mean in Baseline}}{\text{G-Mean in Baseline}} \tag{5.1}$$

The difference in G-Mean gained by a SMOTE variant for each dataset was then averaged over all the 6 datasets.

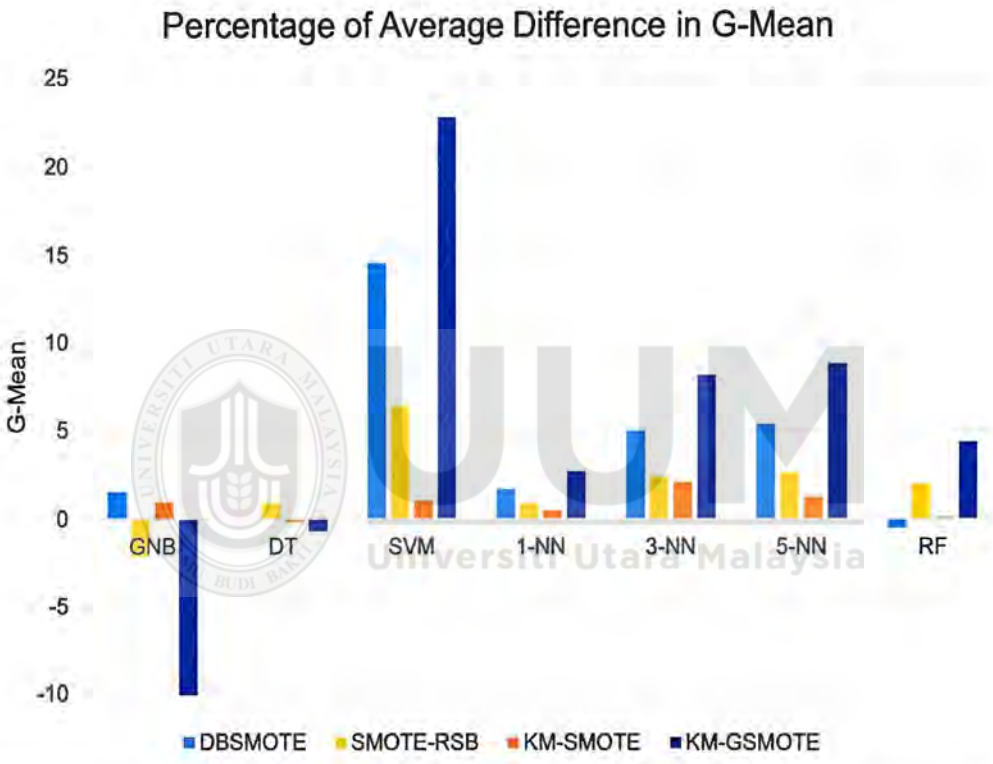


Figure 5.1. Percentage of average difference in G-Mean between SMOTE variants and without SMOTE (baseline).

The following observations can be concluded from the Figure. 5.1:

1. The proposed SMOTE variants, KM-GSMOTE has the highest percentage of average difference of G-Mean for all classifiers including Support Vector Machine (22.76%), 1-Nearest Neighbour (2.66%), 3-Nearest Neighbour (8.14%), 5-Nearest Neighbour (8.80%) and Random Forest (4.35%) except for GNB and DT in which this variants records the lowest percentage, -10.11% and -0.70%

respectively. Despite that, it may be concluded that KM-GSMOTE could generate more informative synthetic examples in comparison with the other SMOTE variants which eventually leads to construction of better classification model by most classifiers.

2. On the other hand, the performance of KM-SMOTE is consistently lower compared to the other SMOTE variants in most classifiers. Particulalry, a huge gap in percentage of average difference values can be noticed between KM-SMOTE and KM-GSMOTE in which the latter increases the G-Mean values significantly than the former by 21.70% in SVM, 2.15% in 1-NN, 6.09% in 3-NN, 7.57% in 5-NN, and 4.24% in RF.
3. From experimental result, SVM has proven as the best classifier for all SMOTE variants. This classifier gives the highest percentage of average difference in G-Mean for all experimented SMOTE variants except for KM-SMOTE with value of 14.56% obtained from DBSMOTE, 6.42% from SMOTE-RSB, and 22.76% from KM-GSMOTE.
4. GNB is observed to have the least significant classification performance followed by DT. GNB produces the lowest percentage for KM-GSMOTE (-10.11%) and SMOTE-RSB* (-2.16%) relative to other classifiers while negative percentage can be frequently found in DT's result; DBSMOTE (-0.04%), KM-SMOTE (-0.13%) and KM-GSMOTE (-0.70%).

Table 5.1

Number of datasets that have classification performance improvement based on G-Mean (out of 6 datasets).

SMOTE Variant	**GNB**	**DT**	**SVM**	**1-NN**	**3-NN**	**5-NN**	**RF**
DBSMOTE (Bunkhumpornpat et al., 2012)	5	4	5	3	4	4	1
SMOTE-RSB* (Ramentol et al., 2012)	0	4	5	4	4	3	2
KM-SMOTE (Douzas et al., 2018)	5	3	5	3	4	3	2
KM-GSMOTE	1	3	5	3	4	4	5

Table 5.1 shows the number of datasets, out of 6 datasets used in the experiments that shows improvement in classification performance with regards to G-Mean for each SMOTE variant and classifier. The observations from the figure can be concluded as follows:

1. DBSMOTE is able to enhance the classification performance where the average difference of G-Mean is more than zero in more than half of the total datasets for most of the classifiers excluding 1-NN and RF. The same finding applies to KM-GSMOTE considering the number of datasets with positive average difference of G-Mean more than 3 can be noticed from SVM, 3-NN, 5-NN and RF. Apparently, this might be the result of the synthetic examples generated in the datasets provide more additional information.
2. Referring to Figure 5.1, the percentage of average difference in G-Mean value for KM-GSMOTE is the lowest in GNB. This is probably because the classification performance has been improved in one dataset only and the other datasets might have been greatly degraded. Even though the number of dataset improved is the same in DBSMOTE and RF while none in SMOTE-RSB and GNB, the difference in G-Mean values are much lower compared to KM-GSMOTE and GNB. The explanation behind this circumstance is that the classification performance in some datasets might remain the same even after oversampling.
3. SVM has improved the classification performance in 5 out of 6 datasets for all SMOTE variants while 3-NN in 4 datasets which imply both of the classifiers are capable of designing a better classification model that has a good predictive ability and could classify the data more correctly. The result for SVM in this table is consistent with the one in Figure 5.1 where SVM clearly exhibit outstanding performances.
4. However, it appears that the classification model trained by GNB could not ac-

curately classify the data which bring about to no improvement gained in any of the experimented datasets.

Next, F1-Score is computed as a metric in evaluating the classifier performance in term of minority class performance rate (t_{pos}) relative to the f_{pos} and f_{neg}.

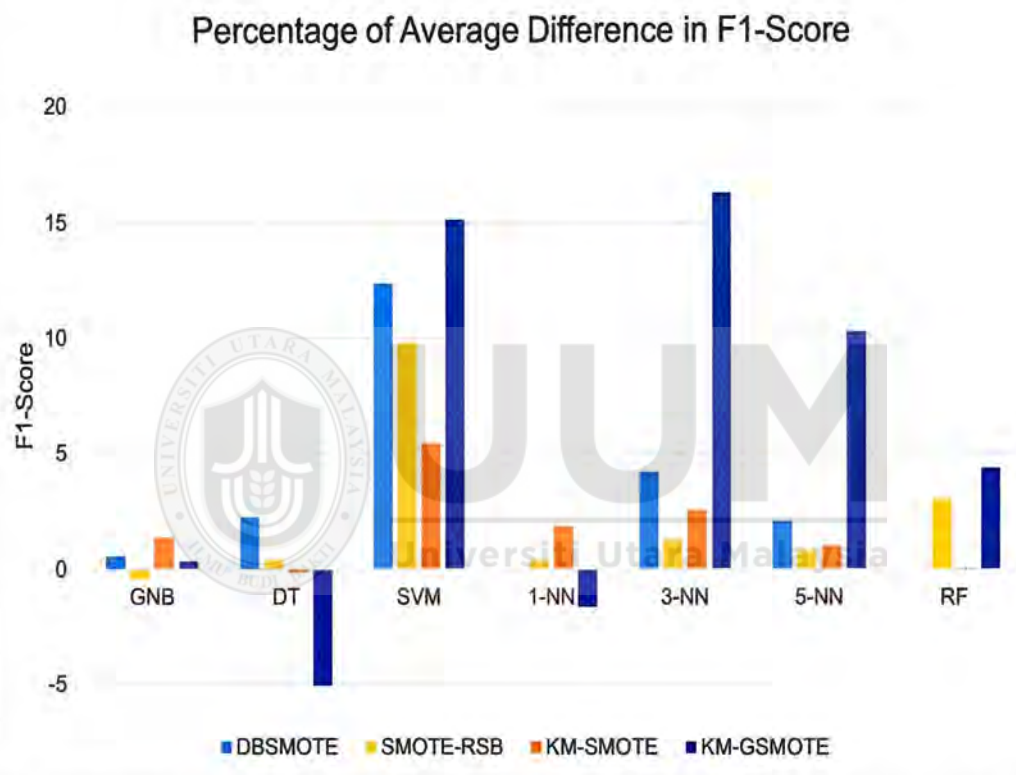


Figure 5.2. Percentage of average difference in F1-Score between SMOTE variants and without SMOTE (baseline).

For each classifier, Figure. 5.2 shows the percentage of average difference in F1-Score between SMOTE variants and without SMOTE. The values were calculated by getting the difference between the F1-Score of each SMOTE variant for each dataset with the F1-Score of without SMOTE (baseline), divided by the baseline, as shown in Equation 5.2.

$$\Delta\text{F1-Score} = \frac{\text{F1-Score in SMOTE Variant} - \text{F1-Score in Baseline}}{\text{F1-Score in Baseline}} \tag{5.2}$$

The total of the differences in F1-Score for each SMOTE variant was then divided by the number of (6) datasets to get the average. Based on Figure. 5.2, the observation concluded were as follows:

1. KM-GSMOTE records the highest percentage of average difference in F1-Score for SVM (15.13%), 3-NN (16.28%), 5-NN (10.29%) and RF (4.36%) in which a wide gap in difference can be detected between the highest and the lowest values. However, KM-GSMOTE is not able to achieve the same performance for DT (-5.14%) and 1-NN (-1.64%) as the results of F1-score after oversampling is substantially lower than before oversampling.
2. SMOTE-RSB and KM-SMOTE have comparable classification performance but the former does notably outperform the latter as noticed in SVM and RF. The same observation as Figure 5.1 can be concluded for this figure in terms of the difference in classification performance between KM-GSMOTE and KM-SMOTE. The difference in SVM is 9.71%, in 3-NN is 13.71%, in 5-NN is 9.26%, and in RF is 4.30%.
3. SVM can comparatively classify data better followed by 3-NN considering the percentage of average difference of F1-Scores. There are 12.38%, 9.82%, 5.42%, and 15.13% increment in DBSMOTE, SMOTE-RSB, KM-SMOTE, and KM-GSMOTE respectively.
4. Poor classification performances are observed in GNB, DT and 1-NN where KM-GSMOTE have negative differences in both DT and 1-NN while SMOTE-RSB has a negative difference in GNB. Nevertheless, DT is distinguished to worsen the classification performance the most given that there are two SMOTE variants with negative percentage of average difference in F1-Score; KM-SMOTE (-0.13%) and KM-GSMOTE (-5.14%).

Table 5.2

Number of datasets that have improvement of classification performance based on F1-Score (out of 6 datasets).

SMOTE Variant	GNB	DT	SVM	1-NN	3-NN	5-NN	RF
DBSMOTE (Bunkhumpornpat et al., 2012)	1	5	4	1	3	3	1
SMOTE-RSB* (Ramentol et al., 2012)	2	4	4	1	1	1	2
KM-SMOTE (Douzas et al., 2018)	1	3	3	1	1	1	1
KM-GSMOTE	3	3	5	4	5	6	5

Table 5.2 presents the number of datasets that have positive average difference of F1-Score for each SMOTE variants and classifiers which explains that more examples are being classified more correctly after oversampling. The following observations can be made:

1. The result of KM-GSMOTE in this table is in accordance with Figure 5.2. KM-GSMOTE shows the ability to improve the classification performance in more than half of the total datasets for most of the classifiers such as SVM, 1-NN, 3-NN, 5-NN, and RF such that the values of F1-score after oversampling is more than the values before oversampling. Exception can be noticed for GNB and DT where KM-GSMOTE can only make improvement in exactly 3 out of 6 datasets. Even though there are 3 datasets that experience better classification performance in DT, the average difference values is negative as illustrated in Figure 5.2 due to the fact that the degradation of performance is more overwhelming.
2. For classifiers, only SVM manages to enhance the classification performance by training a better classification model in more than half of the total datasets except for KM-SMOTE with 3 out 6 datasets.

Figure. 5.3 illustrates the average performance of SMOTE variants with respect to

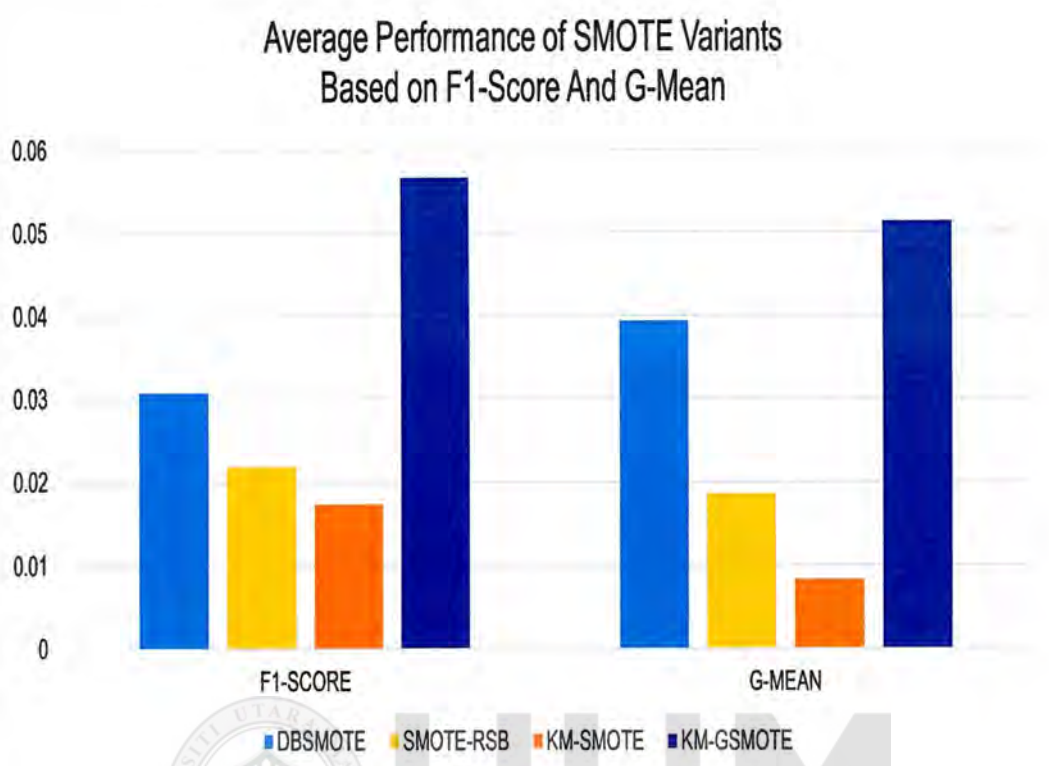


Figure 5.3. Average performance of SMOTE variants with reference to the results of F1-Score and G-Mean

F1-score and G-Mean when combined with different base classifiers. According to the figure, KM-GSMOTE has the best average performance based on both F1-Score (0.057) and G-Mean (0.051). On the contrary, KM-GSMOTE has the lowest values for both F1-Score (0.017) and G-Mean (0.008) which implies that the performance of this technique is barely noticeable.

5.3 SMOTE Variants Complexity Analysis

In brief, higher values of N1 indicates high degree of noise examples exist in the dataset due to N1 complexity being very sensitive to noise examples as well as more examples can be found near the class boundary. While low values of N1 indicates fewer noise examples in the dataset and there is a larger separation in distribution of classes which eventually makes the classification task easier Morán-Fernández et al.

(2017).

Hence, this subsection will discuss the performance of SMOTE variants based on the complexity measure, N1 value as illustrated in Figure 5.4. The column graphs represent the differences of complexity measures after SMOTE variants are applied on all 6 datasets. The difference in N1 complexity is calculated using Equation 5.3.

$$\Delta\text{Complexity} = \frac{\text{Complexity After SMOTE Variant} - \text{Complexity in Baseline}}{\text{Complexity in Baseline}} \quad (5.3)$$

The following observations have been concluded based on this figure:

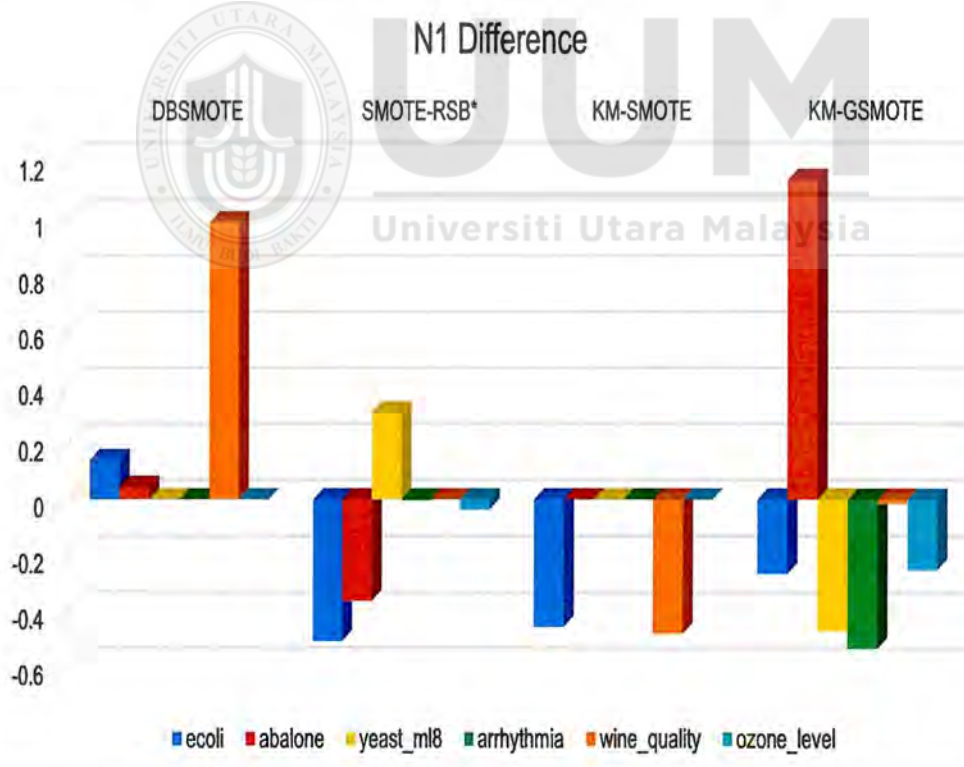


Figure 5.4. The difference in N1 complexity value between before and after implementation of SMOTE variants in all datasets.

1. A negative difference in N1 complexity value suggests that the decision boundary can be defined easily because of there are lesser examples from opposite

classes lay near the boundary and examples from the same class are closely located to each other. This scenario can be observed from 5.4 whereby all SMOTE variants except for DBSMOTE are able to reduce the N1 complexity in a dataset and also the difficulty level of classification task.

2. DBSMOTE has the poorest performance since none of the experimented datasets experience reduction in the complexity of decision boundary, instead the complexity is either increasing or remain unchanged.
3. The performances of SMOTE-RSB* and KM-SMOTE are comparable as negative differences can be observed in *ecoli* dataset for both variants with values of -0.50583 and -0.45374 respectively. SMOTE-RSB* decreases the N1 complexity value in *abalone* and *ozone_level* datasets by -0.36277 and -0.035632 respectively. However, for *yeast_ml8* dataset, the N1 difference is 0.30924 which implies the separation between different classes has been narrower. N1 difference for *wine_quality* dataset after using KM-SMOTE is -0.47930 while for the other datasets, there are no changes in the level of complexity.
4. Overall, KM-GSMOTE shows notable performances in lowering the complexity of decision boundary in all datasets such as *ecoli* (ΔN1 $= -0.26413$), *yeast_ml8* (ΔN1 $= -0.46817$), *arrhythmia* (ΔN1 $= -0.53139$), *wine_quality* (ΔN1 $= -0.01482$), and *ozone_level* (ΔN1 $= -0.25000$) except for *abalone* dataset (ΔN1 $= 1.13705$).

Figure 5.5 and Figure 5.6 describe the distribution of changes in F1-Score and G-mean respectively over the change in N1 complexity, from without SMOTE processing (baseline) to with SMOTE processing. Several observations can be extracted from these figures such as follows:

1. The distribution are scattered in four different quadrants namely quadrant 1, quadrant 2, quadrant 3, and quadrant 4 starting from the top right of the graph

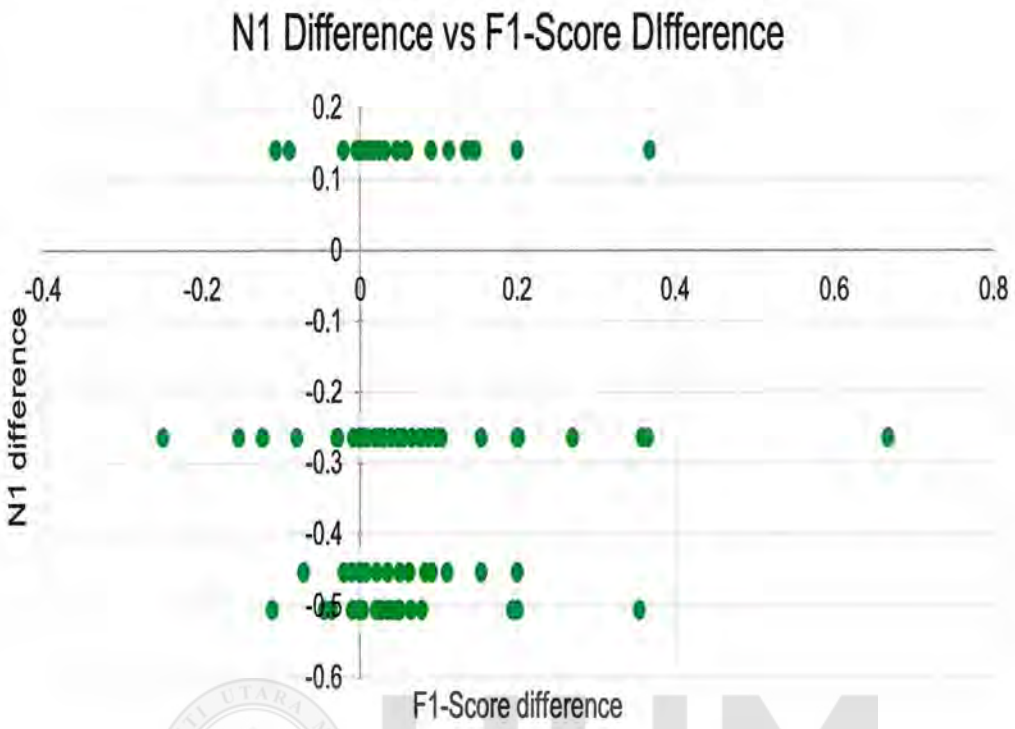


Figure 5.5. The changes in N1 and F1-Score value from classification task without SMOTE processing (baseline) to classification task with SMOTE processing, respectively.

in clockwise order. On the quadrant 1, it is observed that there is a F1-score improvement despite of the value of N1 complexity difference is positive. Nonetheless, only few distribution are located in this quadrant, implying that there is low possibility of getting better classification performance and at the same time, the decision boundary complexity has been exacerbated.

2. The distribution in quadrant 2 represent experimentation results that have performance increment in F1-score metric, while also having reduction in N1 complexity value. Majority of the experimentation results are more likely to experience better classification performance when less complex boundary is needed to separate the classes. The highest ΔF1-score is 0.66667 when the ΔN1 is -0.26413
3. Quadrant 3 describes experimentation results that have performance reduction

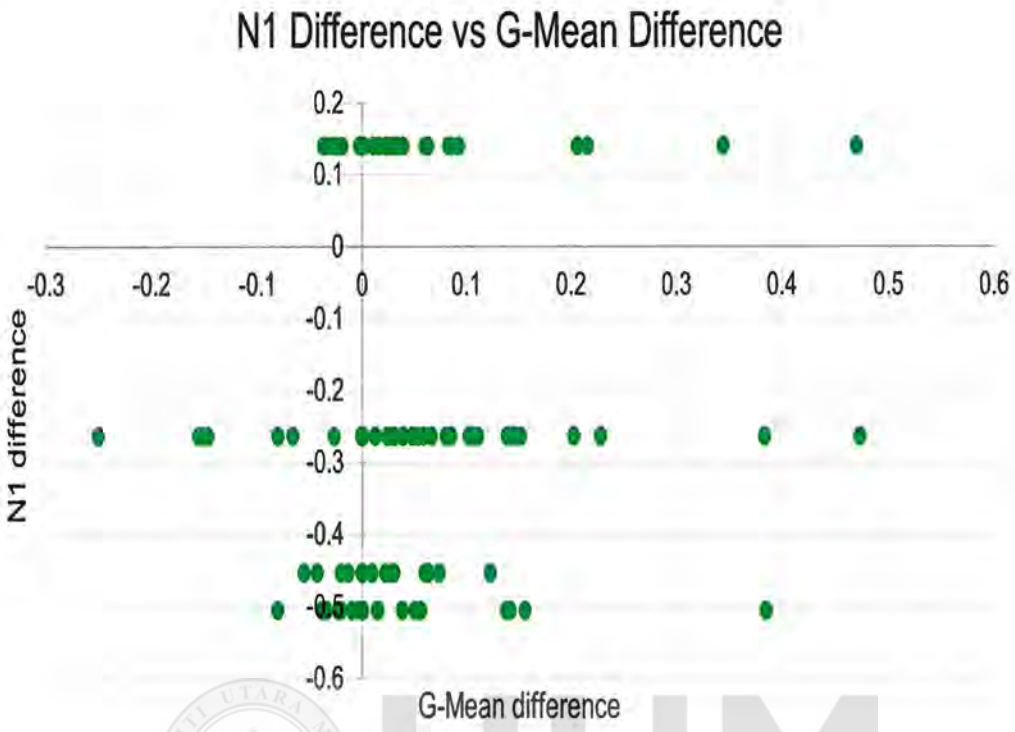


Figure 5.6. The changes in N1 and G-mean value from classification task without SMOTE processing (baseline) to classification task with SMOTE processing, respectively.

in both F1-score metric and N1 complexity value. Some of them suffer from poor classification performance in spite of the decision boundary becoming less complex. From observation, most of the experimentation results in this quadrant have F1 difference value that close to zero which means the degradation of classification performance is insignificant. These few experimentation results are exceptional as many of the of the other results in quadrant 2 are showing promising F1-score values when degradation in N1 complexity values are observed.

4. The distribution in quadrant 4 suggests experimentation results that are have reduction in F1-score performance due to the increase in N1 complexity. Only few experimentation results can be found in this quadrant while most of them that have more complex decision boundary still manage to improve the classification

performance as illustrated in quadrant 1.

5.4 Summary

Section 5.2 indicates that KM-GSMOTE has the best classification performances in comparison with the selected benchmark SMOTE variants which are DBSMOTE, SMOTE-RSB* and KM-SMOTE. KM-GSMOTE has proven that it manages to improve the binary classification performance by generating synthetic minority examples in a geometric region of a truncated hyper-spheroid. KM-GSMOTE has been observed to increase the G-Mean and F1-Score values after oversampling and eventually improves the learning task performed by classifier. The highest percentage of average differences of G-Mean and F1-Score recorded for this technique are 22.76% and 15.13% respectively where both results are obtained for SVM classifier. In addition, KM-GSMOTE also has the best average performance in reference to G-Mean and F1-Score values as compared to the other SMOTE variants.

The experimental results in section 5.3 have proven that there are correlation between performance metrics; F1-Score & G-Mean, N1 complexities and SMOTE variants. The probability of achieving better classification performance has become much higher as a consequences of reduction in intrinsic complexity and the classification task becomes more easier. There are some cases in which the F1-Score or G-Mean values increase after oversampling despite positive N1 difference which implies the complexity of the datasets has been slightly worsen. But generally, a negative N1 difference would result in positive F1-Score and G-Mean differences.

The selection of classification result for proposed SMOTE variant, KM-GSMOTE is made based on the complexity measure. Since this variant incorporates k-mean clustering algorithm, different number of k clusters where k={2,4,6,8} have been tested

for every dataset. Cluster that obtains lower complexity value is selected since it has greater classification performance result.

CHAPTER SIX
CONCLUSION

6.1 Introduction

This chapter concludes the overall proposed KM-GSMOTE. The focuses of this thesis are to discover the correlation between classification performance and intrinsic complexity and to utilize the aforementioned information in KM-GSMOTE algorithm. Section 6.2 will discuss more on the contributions of this research study on proposed KM-GSMOTE in class imbalance problem. Lastly, the future work to enhance SMOTE variant experimentation is also suggested in Section 6.3.

6.2 Contributions of the Research

In the real world, classification task is not a straightforward task. Especially, when dealing with imbalanced dataset, one needs to ensure that performance metric is representing the classification performance of each targeted class. Various strategies have been proposed to solve this issue, in which SMOTE is one of them.

There are several contributions proposed in this study which includes the introduction of a new SMOTE variant known as KM-GSMOTE and discovery of relationship between the classification performance and the degree of data complexity. First objective is to identify the suitable complexity measures that have correlation with performance measures. The experiment measures the N1 data complexity and the classification performance of chosen datasets before and after oversampling takes place. N1 represents the percentage of data points in a minimum spanning tree that are linked to the opposing class by an edge, thus accentuating the traits of the decision boundary between classes. From observation, the chances of getting significant classification task becomes higher when N1 value is low. This implies the definition of decision bound-

ary has become much simpler and relatively easier which enhances the classification model's effectiveness by achieving a balance between recall, precision and specificity, resulting in an improved G-mean an F1-score. The model is able to efficiently capture the characteristics of minority and majority classes while minimizing false negative and false positive respectively.

Second objective is to propose a new SMOTE variant, KM-GSMOTE that integrates data complexity measure as well as model selection that chooses model with a better classification performance and lower data complexity level. During the clustering step, the number of clusters, k range is determined to be 2,4,6 and 8 since it might allow the minority class instances to have a stronger influence on the classification and prevent the majority class from overpowering the decision-making process. The data complexity for every k cluster is observed to discover the differences in N1 complexity value after oversampling takes place. Due to the fact that lower N1 results in better classification performance, k with lower data complexity is chosen to be final result.

Third objective is evaluate the classification performance of KM-GSMOTE. Experiments are conducted on 6 benchmark binary datasets to explain the behaviour of 4 SMOTE variants which are DBSMOTE, SMOTE-RSB*, KM-SMOTE and including the proposed KM-GSMOTE during the classification task. It can be observed that KM-GSMOTE achieves the highest performance gain in the classification metrics G-Mean and F1-Score, which are up to about 23% and 17%, respectively compared to the other SMOTE variants. This is due to the use of k-means clustering, which has a wide range of practical applications, and G- SMOTE, which is often an improved version of SMOTE that addresses the shortcomings of the SMOTE data generation system by extending the range of the linear interpolation mechanism by defining a geometric region. Experiment conducted by Douzas et al. (2019) implements G-SMOTE

in detection of land cover change and rare land cover type classification. The result has proven that this technique yields better G-Mean and F1-Score values compared the other SMOTE variants. The classification performance of G-SMOTE is further amplified in this study via application of k-means clustering and data complexity analysis.

Apart from that, a taxonomy of SMOTE variants is also discussed to best highlight the similarities and differences among the SMOTE variants proposed in the literature. The compared dimensions are based on SMOTE algorithm and which data complexity the SMOTE variant is solving.

6.3 Future Work

For further study, this study would like to propose investigation of other data complexity measures that can be used to benchmark and correlate the SMOTE behaviour in relation to classification performance. In addition to that, identification of the proper complexity measure for each type of problem that is discussed in this study, which is the noise example problem, the class overlap/separability problem and the small disjuncts problem. Future SMOTE variant experimentation should attempt on decreasing the complexity of a dataset due to higher probability of getting good classification performance. To do that, three guidelines are suggested as follows:

1. Generation of synthetic examples in the region with high concentration of minority examples should be avoided since it does not guarantee that the complexity could be reduced. Thus, the actual performance of such a variant might not be any better.
2. The oversampling strategy should incorporate the examples located near the overlapping region as well as working on the the separability between different classes. Not only the required decision boundary would be then less complex

but also the synthetic examples generated could provide effective information that might enhance the classification task.

3. Filtering-based algorithm should have been widely implemented with SMOTE since SMOTE variants that use this type of combination could positively affect the classification performance. This is due to the degree of complexity in a dataset that would subside.

REFERENCES

Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., and Abuzneid, A. (2019). Features dimensionality reduction approaches for machine learning based network intrusion detection. *Electronics*, 8(3):322.

Adedigba, A. P., Adeshina, S. A., Aina, O. E., and Aibinu, A. M. (2021). Optimal hyperparameter selection of deep learning models for covid-19 chest x-ray classification. *Intelligence-based medicine*, 5:100034.

Adhikari, S., Thapa, S., and Shah, B. K. (2020). Oversampling based classifiers for categorization of radar returns from the ionosphere. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 975–978. IEEE.

Agrawal, K., Baweja, Y., Dwivedi, D., Saha, R., Prasad, P., Agrawal, S., Kapoor, S., Chaturvedi, P., Mali, N., Kala, V. U., et al. (2017). A comparison of class imbalance techniques for real-world landslide predictions. In *2017 International Conference on Machine Learning and Data Science (MLDS)*, pages 1–8. IEEE.

Al Helal, M., Haydar, M. S., and Mostafa, S. A. M. (2016). Algorithms efficiency measurement on imbalanced data using geometric mean and cross validation. In *2016 international workshop on computational intelligence (IWCI)*, pages 110–114. IEEE.

Al Majzoub, H., Elgedawy, I., Akaydın, Ö., and Ulukök, M. K. (2020). Hcab-smote: A hybrid clustered affinitive borderline smote approach for imbalanced data binary classification. *Arabian Journal for Science and Engineering*, pages 1–18.

Alashwal, H., El Halaby, M., Crouse, J. J., Abdalla, A., and Moustafa, A. A. (2019). The application of unsupervised clustering methods to alzheimer's disease. *Frontiers in computational neuroscience*, 13:31.

Ali, H., Salleh, M. N. M., Hussain, K., Ahmad, A., Ullah, A., Muhammad, A., Naseem, R., and Khan, M. (2019a). A review on data preprocessing methods for class imbalance problem. *International Journal of Engineering & Technology*, 8:390–397.

Ali, N., Neagu, D., and Trundle, P. (2019b). Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SN Applied Sciences*, 1:1–15.

Almutairi, W. and Janicki, R. (2020). On relationships between imbalance and overlapping of datasets. In *CATA*, pages 141–150.

Apuke, O. D. (2017). Quantitative research methods: A synopsis approach. *Kuwait Chapter of Arabian Journal of Business and Management Review*, 33(5471):1–8.

Arafa, A., El-Fishawy, N., Badawy, M., and Radad, M. (2022). Rn-smote: Reduced noise smote based on dbscan for enhancing imbalanced data classification. *Journal of King Saud University-Computer and Information Sciences*, 34(8):5059–5074.

Barella, V. H., Garcia, L. P., de Souto, M. C., Lorena, A. C., and de Carvalho, A. C. (2021). Assessing the data complexity of imbalanced datasets. *Information Sciences*, 553:83–109.

Barella, V. H., Garcia, L. P., de Souto, M. P., Lorena, A. C., and de Carvalho, A. (2018). Data complexity measures for imbalanced classification tasks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Barua, S., Islam, M. M., Yao, X., and Murase, K. (2012). Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on knowledge and data engineering*, 26(2):405–425.

Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29.

Bennin, K. E., Keung, J., Phannachitta, P., Monden, A., and Mensah, S. (2017). Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. *IEEE Transactions on Software Engineering*, 44(6):534–550.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., et al. (2013). Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*.

Bunkhumpornpat, C. and Sinapiromsaran, K. (2017). Dbmute: density-based majority under-sampling technique. *Knowledge and Information Systems*, 50(3):827–850.

Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2009). 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*, pages 475–482. Springer.

Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2012). Dbsmote: density-based synthetic minority over-sampling technique. *Applied Intelligence*, 36(3):664–684.

Cervantes, J., Garcia-Lamont, F., Rodriguez, L., López, A., Castilla, J. R., and Trueba, A. (2017). Pso-based method for svm classification on skewed data sets. *Neurocomputing*, 228:187–197.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, B., Xia, S., Chen, Z., Wang, B., and Wang, G. (2020). Rsmote: A self-adaptive robust smote for imbalanced problems with label noise. *Information Sciences*.

Chen, B., Xia, S., Chen, Z., Wang, B., and Wang, G. (2021). Rsmote: A self-adaptive robust smote for imbalanced problems with label noise. *Information Sciences*, 553:397–428.

Chen, X., Kang, Q., Zhou, M., and Wei, Z. (2016). A novel under-sampling algorithm based on iterative-partitioning filters for imbalanced classification. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 490–494. IEEE.

Cheng, K., Zhang, C., Yu, H., Yang, X., Zou, H., and Gao, S. (2019a). Grouped smote with noise filtering mechanism for classifying imbalanced data. *IEEE Access*, 7:170668–170681.

Cheng, R., Mamoulis, N., Sun, Y., and Huang, X. (2019b). *Web Information Systems Engineering–WISE 2019: 20th International Conference, Hong Kong, China, January 19–22, 2020, Proceedings*, volume 11881. Springer Nature.

Cieslak, D. A., Chawla, N. V., and Striegel, A. (2006). Combating imbalance in network intrusion datasets. In *GrC*, pages 732–737. Citeseer.

Darwish, S. M. (2020). An intelligent credit card fraud detection approach based on semantic fusion of two classifiers. *Soft Computing*, 24(2):1243–1253.

Das, S., Datta, S., and Chaudhuri, B. B. (2018). Handling data irregularities in classification: Foundations, trends, and future challenges. *Pattern Recognition*, 81:674–693.

Datta, S., Nag, S., Mullick, S. S., and Das, S. (2017). Diversifying support vector machines for boosting using kernel perturbation: Applications to class imbalance and small disjuncts. *arXiv preprint arXiv:1712.08493*.

Douzas, G. and Bacao, F. (2017). Self-organizing map oversampling (somo) for imbalanced data set learning. *Expert systems with Applications*, 82:40–52.

Douzas, G. and Bacao, F. (2019). Geometric smote a geometrically enhanced drop-in replacement for smote. *Information sciences*, 501:118–135.

Douzas, G., Bacao, F., Fonseca, J., and Khudinyan, M. (2019). Imbalanced learning in land cover classification: Improving minority classes' prediction accuracy using the geometric smote algorithm. *Remote Sensing*, 11(24):3040.

Douzas, G., Bacao, F., and Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, 465:1–20.

Douzas, G., Rauch, R., and Bacao, F. (2021). G-somo: An oversampling approach based on self-organized maps and geometric smote. *Expert Systems with Applications*, 183:115230.

Du, G., Zhang, J., Luo, Z., Ma, F., Ma, L., and Li, S. (2020). Joint imbalanced classification and feature selection for hospital readmissions. *Knowledge-Based Systems*, 200:106020.

Faris, H., Abukhurma, R., Almanaseer, W., Saadeh, M., Mora, A. M., Castillo, P. A., and Aljarah, I. (2020). Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the spanish market. *Progress in Artificial Intelligence*, 9(1):31–53.

Fernandes, E. R. and de Carvalho, A. C. (2019). Evolutionary inversion of class distribution in overlapping areas for multi-class imbalanced learning. *Information Sciences*, 494:141–154.

Fernández, A., Garcia, S., Herrera, F., and Chawla, N. V. (2018). Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905.

Fonseca, J., Douzas, G., and Bacao, F. (2021). Improving imbalanced land cover classification with k-means smote: Detecting and oversampling distinctive minority spectral signatures. *Information*, 12(7):266.

Gao, X., Ren, B., Zhang, H., Sun, B., Li, J., Xu, J., He, Y., and Li, K. (2020a). An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling. *Expert Systems with Applications*, 160:113660.

Gao, Y., Li, Y.-F., Lin, Y., Aggarwal, C., and Khan, L. (2020b). Setconv: A new approach for learning from imbalanced data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1284–1294.

Garcia, L. P., de Carvalho, A. C., and Lorena, A. C. (2015). Effect of label noise in the complexity of classification problems. *Neurocomputing*, 160:108–119.

García, V., Sánchez, J. S., Domínguez, H. O., and Cleofas-Sánchez, L. (2015). Dissimilarity-based learning from imbalanced data with small disjuncts and noise. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 370–378. Springer.

Govender, P. and Sivakumar, V. (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research*, 11(1):40–56.

Grandini, M., Bagli, E., and Visani, G. (2022). Metrics for multi-class classification: an overview. arxiv preprint. 2020: 1–17.

Guan, H., Zhang, Y., Xian, M., Cheng, H., and Tang, X. (2020). Smote-wenn: Solving class imbalance and small sample problems by oversampling and distance scaling. *Applied Intelligence*, pages 1–16.

Guzmán-Ponce, A., Valdovinos, R. M., Sánchez, J. S., and Marcial-Romero, J. R. (2020). A new under-sampling method to face class overlap and imbalance. *Applied Sciences*, 10(15):5164.

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

Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer.

Ho, T. K. and Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):289–300.

Jiao, Y. and Du, P. (2016). Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology*, 4:320–330.

Ju, L., Wang, X., Wang, L., Mahapatra, D., Zhao, X., Harandi, M., Drummond, T., Liu, T., and Ge, Z. (2021). Improving medical image classification with label noise using dual-uncertainty estimation. *arXiv preprint arXiv:2103.00528*.

Kansal, T., Bahuguna, S., Singh, V., and Choudhury, T. (2018). Customer segmentation using k-means clustering. In *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)*, pages 135–139. IEEE.

Kaur, P. and Gosain, A. (2018). Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In *ICT Based Innovations*, pages 23–30. Springer.

Kong, J., Kowalczyk, W., Nguyen, D. A., Bäck, T., and Menzel, S. (2019). Hyperparameter optimisation for improving classification under class imbalance. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 3072–3078. IEEE.

Kovács, G. (2019a). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, 83:105662.

Kovács, G. (2019b). Smote-variants: A python implementation of 85 minority oversampling techniques. *Neurocomputing*, 366:352–354.

Koziarski, M., Krawczyk, B., and Woźniak, M. (2019). Radial-based oversampling for noisy imbalanced data classification. *Neurocomputing*, 343:19–33.

Koziarski, M., Woźniak, M., and Krawczyk, B. (2020). Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise. *Knowledge-Based Systems*, 204:106223.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

Kunakorntum, I., Hinthong, W., and Phunchongharn, P. (2020). A synthetic minority based on probabilistic distribution (symprod) oversampling for imbalanced datasets. *IEEE Access*, 8:114692–114704.

Larrazabal, A. J., Nieto, N., Peterson, V., Milone, D. H., and Ferrante, E. (2020). Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proceedings of the National Academy of Sciences*, 117(23):12592–12594.

Le, T., Vo, M. T., Vo, B., Lee, M. Y., and Baik, S. W. (2019). A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction. *Complexity*, 2019.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42.

Li, B., Wang, Y., Singh, A., and Vorobeychik, Y. (2016). Data poisoning attacks on factorization-based collaborative filtering. *arXiv preprint arXiv:1608.08182*.

Liang, X., Jiang, A., Li, T., Xue, Y., and Wang, G. (2020). Lr-smote–an improved unbalanced data set oversampling based on k-means and svm. *Knowledge-Based Systems*, page 105845.

Lin, W.-C., Tsai, C.-F., Hu, Y.-H., and Jhang, J.-S. (2017). Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409:17–26.

Liu, C., Jin, S., Wang, D., Luo, Z., Yu, J., Zhou, B., and Yang, C. (2020). Constrained oversampling: An oversampling approach to reduce noise generation in imbalanced datasets with class overlapping. *IEEE Access*.

Lorena, A. C., Garcia, L. P., Lehmann, J., Souto, M. C., and Ho, T. K. (2019). How complex is your classification problem? a survey on measuring classification complexity. *ACM Computing Surveys (CSUR)*, 52(5):1–34.

Lu, Y., Cheung, Y.-M., and Tang, Y. Y. (2019). Bayes imbalance impact index: A measure of class imbalanced data set for classification problem. *IEEE transactions on neural networks and learning systems*, 31(9):3525–3539.

Maldonado, S., López, J., and Vairetti, C. (2019). An alternative smote oversampling strategy for high-dimensional datasets. *Applied Soft Computing*, 76:380–389.

Maulidevi, N. U., Surendro, K., et al. (2021). Smote-lof for noise identification in imbalanced data classification. *Journal of King Saud University-Computer and Information Sciences*.

Mohammed, A. J., Hassan, M. M., and Kadir, D. H. (2020). Improving classification performance for a novel imbalanced medical dataset using smote method. *International Journal*, 9(3).

Morán-Fernández, L., Bolón-Canedo, V., and Alonso-Betanzos, A. (2017). Can classification performance be predicted by complexity measures? a study using microarray data. *Knowledge and Information Systems*, 51(3):1067–1090.

Mullick, S. S., Datta, S., and Das, S. (2019). Generative adversarial minority oversampling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1695–1704.

Napierala, K. and Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems*, 46(3):563–597.

Nekooeimehr, I. and Lai-Yuen, S. K. (2016). Adaptive semi-unsupervised weighted oversampling (a-suwo) for imbalanced datasets. *Expert Systems with Applications*, 46:405–416.

Nguyen, P. T., Ha, D. H., Avand, M., Jaafari, A., Nguyen, H. D., Al-Ansari, N., Van Phong, T., Sharma, R., Kumar, R., Le, H. V., et al. (2020). Soft computing ensemble models based on logistic regression for groundwater potential mapping. *Applied Sciences*, 10(7):2469.

Nguyen, Q. H., Ly, H.-B., Ho, L. S., Al-Ansari, N., Le, H. V., Tran, V. Q., Prakash, I., and Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, 2021:1–15.

Nigam, N., Dutta, T., and Gupta, H. P. (2020). Impact of noisy labels in learning techniques: a survey. In *Advances in data and information sciences*, pages 403–411. Springer.

Parambath, S. P., Usunier, N., and Grandvalet, Y. (2014). Optimizing f-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems*, pages 2123–2131.

Patel, H., Singh Rajput, D., Thippa Reddy, G., Iwendi, C., Kashif Bashir, A., and Jo, O. (2020). A review on classification of imbalanced data for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 16(4):1550147720916404.

Pelletier, C., Valero, S., Inglada, J., Champion, N., Marais Sicre, C., and Dedieu, G. (2017). Effect of training class label noise on classification performances for land cover mapping with satellite image time series. *Remote Sensing*, 9(2):173.

Piri, S., Delen, D., and Liu, T. (2018). A synthetic informative minority over-sampling (simo) algorithm leveraging support vector machine to enhance learning from imbalanced datasets. *Decision Support Systems*, 106:15–29.

Pozi, M. S. M., Azhar, N. A., Raziff, A. R. A., and Ajrina, L. H. (2022). Svgpm: Evolving svm decision function by using genetic programming to solve imbalanced classification problem. *Prog. in Artif. Intell.*, 11(1):65–77.

Pozi, M. S. M., Sulaiman, M. N., Mustapha, N., and Perumal, T. (2016). Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming. *Neural Processing Letters*, 44(2):279–290.

Ramentol, E., Caballero, Y., Bello, R., and Herrera, F. (2012). Smote-rs b*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowledge and information systems*, 33(2):245–265.

Richhariya, B. and Tanveer, M. (2020). A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognition*, 102:107150.

Rivera, W. A. (2017). Noise reduction a priori synthetic over-sampling for class imbalanced data sets. *Information Sciences*, 408:146–161.

Sáez, J. A., Luengo, J., Stefanowski, J., and Herrera, F. (2015). Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a resampling method with filtering. *Information Sciences*, 291:184–203.

Sanguanmak, Y. and Hanskunatai, A. (2016). Dbsm: The combination of dbscan and smote for imbalanced data classification. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5. IEEE.

Santos, M. S., Abreu, P. H., García-Laencina, P. J., Simão, A., and Carvalho, A. (2015). A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. *Journal of biomedical informatics*, 58:49–59.

Santoso, B., Wijayanto, H., Notodiputro, K., and Sartono, B. (2017). Synthetic over sampling methods for handling class imbalanced problems: a review. In *IOP conference series: earth and environmental science*, volume 58, page 012031.

Sasada, T., Liu, Z., Baba, T., Hatano, K., and Kimura, Y. (2020). A resampling method for imbalanced datasets considering noise and overlap. *Procedia Computer Science*, 176:420–429.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.

Sinaga, K. P. and Yang, M.-S. (2020). Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727.

Soltanzadeh, P. and Hashemzadeh, M. (2021). Rcsmote: Range-controlled synthetic minority over-sampling technique for handling the class imbalance problem. *Information Sciences*, 542:92–111.

Sowah, R. A., Agebure, M. A., Mills, G. A., Koumadi, K. M., and Fiawoo, S. Y. (2016). New cluster undersampling technique for class imbalance learning. *International Journal of Machine Learning and Computing*, 6(3):205.

Spelmen, V. S. and Porkodi, R. (2018). A review on handling imbalanced data. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–11. IEEE.

Tang, Y., Zhang, Y.-Q., Chawla, N. V., and Krasser, S. (2008). Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288.

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., and Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, 7(1):1–47.

Tao, X., Li, Q., Guo, W., Ren, C., He, Q., Liu, R., and Zou, J. (2020). Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering. *Information Sciences*, 519:43–73.

Van Dao, D., Jaafari, A., Bayat, M., Mafi-Gholami, D., Qi, C., Moayedi, H., Van Phong, T., Ly, H.-B., Le, T.-T., Trinh, P. T., et al. (2020). A spatially explicit deep learning neural network model for the prediction of landslide susceptibility. *Catena*, 188:104451.

Verbiest, N. (2014). *Fuzzy rough and evolutionary approaches to instance selection*. PhD thesis, Ghent University.

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

Vuttipittayamongkol, P. and Elyan, E. (2020). Improved overlap-based undersampling for imbalanced dataset classification with application to epilepsy and parkinson's disease. *International journal of neural systems*, 30(08):2050043.

Vuttipittayamongkol, P., Elyan, E., and Petrovski, A. (2020). On the class overlap problem in imbalanced data classification. *Knowledge-based systems*, page 106631.

Wang, S., Minku, L. L., and Yao, X. (2014). Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1356–1368.

Wang, X., Yu, B., Ma, A., Chen, C., Liu, B., and Ma, Q. (2019). Protein–protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. *Bioinformatics*, 35(14):2395–2402.

Wang, Z. and Wang, H. (2021). Global data distribution weighted synthetic oversampling technique for imbalanced learning. *IEEE Access*, 9:44770–44783.

Wei, J., Huang, H., Yao, L., Hu, Y., Fan, Q., and Huang, D. (2020a). Ia-suwo: An improving adaptive semi-unsupervised weighted oversampling for imbalanced classification problems. *Knowledge-Based Systems*, 203:106116.

Wei, J., Huang, H., Yao, L., Hu, Y., Fan, Q., and Huang, D. (2020b). New imbalanced fault diagnosis framework based on cluster-mwmote and mfo-optimized ls-svm using limited and complex bearing data. *Engineering Applications of Artificial Intelligence*, 96:103966.

Wei, J., Huang, H., Yao, L., Hu, Y., Fan, Q., and Huang, D. (2020c). Ni-mwmote: An improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems. *Expert Systems with Applications*, page 113504.

Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp? In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pages 1–6. IEEE.

Wu, C., Yan, B., Yu, R., Yu, B., Zhou, X., Yu, Y., and Chen, N. (2021). k-means clustering algorithm and its simulation based on distributed computing platform. *Complexity*, 2021.

Yan, Y., Liu, R., Ding, Z., Du, X., Chen, J., and Zhang, Y. (2019). A parameter-free cleaning method for smote in imbalanced classification. *IEEE Access*, 7:23537–23548.

Yuan, B.-W., Luo, X.-G., Zhang, Z.-L., Yu, Y., Huo, H.-W., Johannes, T., and Zou, X.-D. (2020). A novel density-based adaptive k nearest neighbor method for dealing with overlapping problem in imbalanced datasets. *Neural Computing and Applications*, pages 1–25.

Yun, J., Ha, J., and Lee, J.-S. (2016). Automatic determination of neighborhood size in smote. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, pages 1–8.

Zhang, G., Zhang, C., and Zhang, H. (2018). Improved k-means algorithm based on density canopy. *Knowledge-based systems*, 145:289–297.

Zhao, W.-L., Deng, C.-H., and Ngo, C.-W. (2018). k-means: A revisit. *Neurocomputing*, 291:195–206.

Zheng, X. (2020). *SMOTE Variants for Imbalanced Binary Classification: Heart Disease Prediction*. PhD thesis, UCLA.

Zhu, T., Lin, Y., and Liu, Y. (2017). Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognition*, 72:327–340.