New Developments for Addressing Class Imbalance Issue in Classification Tasks

By
SURANI MATHARAARACHCHI

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Statistics
University of Manitoba
Winnipeg, Manitoba

Copyright © 2024 by Surani Matharaarachchi

DOCTOR OF PHILOSOPHY (2025)
Statistics
University of Manitoba
Winnipeg, Manitoba, Canada

TITLE: New Developments for Addressing Class Imbalance Issue in Classification Tasks

AUTHOR:

Surani Matharaarachchi,

SUPERVISOR:

Dr. Saman Muthukumarana
Professor, Department of Statistics,
University of Manitoba, MB, Canada

CO-SUPERVISOR:

Dr. Mike Domaratzki,
Professor, Department of Computer Science,
Western University, ON, Canada

SUPERVISORY COMMITTEE MEMBERS:

Dr. Max Turgeon
Professor, Department of Statistics,
University of Manitoba, MB, Canada

Dr. Alan Katz
Professor, Department of Community Health Sciences and Family Medicine,
University of Manitoba, MB, Canada

EXTERNAL EXAMINER:

Dr. Matthew Pratola
Professor, Department of Statistics,
Indiana University, Indiana, USA

NUMBER OF PAGES: xviii, 206

Abstract

Imbalanced datasets pose a significant challenge in machine learning, often resulting in biased models that inadequately learn from minority class instances. The Synthetic Minority Over-sampling Technique (SMOTE) is a well-established method to address this issue by generating synthetic samples for underrepresented classes. This Ph.D. thesis introduces several novel extensions to SMOTE, including Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE, BGMM SMOTE, and Deep ExtSMOTE, each designed to enhance performance and manage complexities in various applications.

Class imbalance, compounded by abnormal instances and the curse of dimensionality, adversely impacts model accuracy and reliability. Many existing techniques, including SMOTE, are insufficient for effectively handling these complexities. Our research develops new methodologies to tackle these challenges. Distance-ExtSMOTE, Dirichlet-ExtSMOTE, FCRP-SMOTE, and BGMM-SMOTE use sophisticated techniques to handle abnormal instances and improve the quality of synthetic samples by leveraging weighted averages of neighbouring data points. These methods effectively manage outliers and noisy data, enhancing the robustness of the classification models. Additionally, DeepExt-SMOTE integrates these methods with autoencoders to further enhance accuracy when dealing with high-dimensional data.

The development of these novel methodologies was inspired by the complexities and class imbalance issue observed in the predictive analysis of long COVID. The challenges encountered in this context highlighted the need for more advanced techniques, underscoring the importance of developing robust solutions for imbalanced datasets.

Empirical studies validate the effectiveness of these extensions, demonstrating significant improvements in classification performance across a range of metrics, including F1 scores, PR-AUC, and MCC. These advancements are particularly relevant in real-world applications, such as medical diagnosis, fraud detection, churn prediction, and fault detection. Addressing class imbalance with abnormal instances through these methods leads to more accurate and reliable predictions,

ultimately contributing to better decision-making and improved outcomes in these critical areas.

This work offers valuable tools for researchers and practitioners by advancing methodologies for handling class imbalance, abnormal instances, and high-dimensional data. The proposed techniques provide enhanced capabilities for managing complex classification tasks and achieving impactful results across diverse application domains.

Keywords:
Class Imbalance
SMOTE
Dirichlet Distribution
Dirichlet Process
Bayesian Gaussian Mixture Models
Dimension Reduction
Auto-encoders
Abnormal Instances

Acknowledgements

First and foremost, I want to express my deepest gratitude to my advisors, Dr. Saman Muthukumarana and Dr. Mike Domaratzki. Your exceptional guidance, steadfast support, and unwavering confidence in my abilities have been the foundation of this thesis. Your insightful feedback and persistent encouragement have not only shaped this work but also profoundly influenced my academic growth. I consider myself incredibly fortunate to have had the privilege of your mentorship throughout my graduate journey; your contributions have been truly invaluable.

I am also very thankful to my advisory committee members, Dr. Max Turgeon and Dr. Alan Katz, for their valuable advice and constructive critiques, which have significantly shaped the direction and quality of this work. Your dedication to my academic growth has been truly inspiring.

I also sincerely appreciate the time and effort the external examiner, Dr. Matthew Pratola dedicated to evaluating my research.

Many thanks to the staff, support staff, and colleagues in the Department of Statistics, University of Manitoba. Additionally, I express my gratitude to the Faculty of Graduate Studies for the financial support through the University of Manitoba Graduate Fellowship (UMGF).

I owe a deep debt of gratitude to the free education system in Sri Lanka, which has provided me with invaluable opportunities and resources to pursue my dreams.

Finally, I want to express my profound gratitude to my family including my loving husband Sudesh, my parents, grandparents, father-in-law, brothers, uncles, aunts, my friends and all my teachers. Their unwavering support and guidance have been instrumental in my journey to this remarkable achievement.

Dedication Page

This thesis is dedicated to my loving husband, Sudesh Chathuranga, who has been my pillar of strength, supporting me with resilience, patience, and unwavering motivation throughout this entire journey. I am truly blessed to have you in my life. I love you so much!

Contents

Abstract **iv**

Acknowledgements **vi**

Dedication Page **vii**

List of Figures **xiii**

List of Tables **xvii**

1 Introduction **1**

1.1 Class Imbalance 1

1.1.1 Challenges in Handling Imbalanced Data 3

1.2 Abnormal Instances 4

1.3 Motivating Examples 5

1.3.1 Identifying Long COVID Patients: Challenges of Imbalanced Data 10

1.4 Objective of the Thesis 10

1.5 Main Contributions and Thesis Organization 11

1.6 Contributions of Authors 13

Bibliography 14

2 Background **20**

2.1 Current Approaches 20

2.1.1 SMOTE 22

2.1.2 Other SMOTE Extensions 23

2.2 Local Outlier Factor (LOF) 24

2.3 Dimensionality Reduction 26

2.3.1 The Curse of Dimensionality 26

2.3.2 Principal Component Analysis (PCA) 27

2.3.3 Autoencoders . . . 27
2.4 Important Classifiers . . . 28
2.4.1 Logistic Regression . . . 28
2.4.2 Logistic Regression with Elastic Net Regularization . . . 29
2.4.3 k-Nearest Neighbors (k-NN) Classifier . . . 30
2.4.4 Random Forest Classifier . . . 31
2.5 Performance Measures . . . 33
Bibliography . . . 35

3 Discovering Long COVID Symptom Patterns: Association Rule Mining and Sentiment Analysis in Social Media Tweets 41
3.1 Introduction . . . 44
3.2 Methods and Experimental Design . . . 45
3.2.1 Data Collection . . . 45
3.2.2 Data Pre-Processing . . . 48
3.2.3 Sentiment Analysis . . . 51
3.2.4 Collocations . . . 52
3.2.5 Association Rule Mining (ARM) . . . 55
3.3 Statistical Analysis and Results . . . 56
3.3.1 Symptom Rules . . . 59
3.4 Discussion . . . 62
3.4.1 Principal Findings . . . 62
3.4.2 Limitations . . . 63
3.4.3 Conclusion . . . 64
3.4.4 Acknowledgments . . . 64
3.4.5 Conflicts of Interest . . . 64
Bibliography . . . 64

4 Long COVID Prediction in Manitoba Using Clinical Notes Data: A Machine Learning Approach 67
4.1 Introduction . . . 69
4.1.1 Related Work . . . 69
4.2 Methods and Experimental Design . . . 71
4.2.1 Data Collection . . . 71
4.2.2 Study Sample . . . 72
4.2.3 Identifying the Known LCS Group . . . 72

4.2.4 Identifying the Control Group . . . 73
4.2.5 Symptom Extraction and Negation Identification . . . 76
4.2.6 Machine Learning Approach . . . 76
4.2.6.1 Class Imbalance . . . 77
4.2.6.2 Logistic Regression . . . 78
4.2.6.3 Logistic Regression with Elastic Net Regularization for Classification . . . 78
4.2.6.4 Random Forest Classification . . . 79
4.2.6.5 Accuracy Measures . . . 80
4.3 Statistical Analysis and Results . . . 81
4.4 Discussion . . . 87
4.5 Conclusion . . . 88
Bibliography . . . 88

5 Enhancing SMOTE for Imbalanced Data with Abnormal Minority Instances 92
5.1 Introduction . . . 94
5.2 Literature Review . . . 95
5.3 Methodology . . . 98
5.3.1 Abnormal Instances . . . 98
5.3.2 Overview of SMOTE . . . 100
5.3.3 Proposed Methods - Based on Weighted Average . . . 101
5.3.4 The Distance Measure . . . 102
5.3.5 Distance ExtSMOTE . . . 103
5.3.6 Dirichlet ExtSMOTE . . . 105
5.3.6.1 Dirichlet Distribution . . . 105
5.3.7 FCRP SMOTE . . . 108
5.3.8 BGMM SMOTE . . . 111
5.3.9 Bayesian Gaussian Mixture Models with EM Algorithm . . . 111
5.3.10 Role of Weighting Mechanism in Mitigating Abnormal Instances . . . 115
5.3.11 Performance Measures . . . 116
5.4 Simulation Results . . . 116
5.4.1 Synthetic Data Generation . . . 116
5.4.2 Simulation Setup . . . 118

5.4.3 Sensitivity Analysis of the Parameters . . . 121
5.5 Application Results . . . 122
5.5.1 Datasets Description . . . 122
5.5.2 Experimental Settings . . . 123
5.5.2.1 Statistical Analysis . . . 125
5.6 Discussion . . . 132
5.6.1 Practical Implications . . . 133
5.6.2 Theoretical Foundations of Proposed Methods . . . 134
5.6.2.1 Distance ExtSMOTE . . . 134
5.6.2.2 Dirichlet ExtSMOTE . . . 134
5.6.2.3 FCRP SMOTE . . . 135
5.6.2.4 BGMM SMOTE . . . 136
5.6.3 Future Research Directions . . . 136
5.7 Conclusion . . . 137
Bibliography . . . 137

6 Deep-ExtSMOTE: Integrating Autoencoders for Advanced Mitigation of Class Imbalance in High-Dimensional Data Classification 143
6.1 Introduction . . . 145
6.2 Methodology . . . 147
6.2.1 Autoencoders . . . 147
6.2.2 SMOTE with Autoencoders . . . 149
6.2.3 Deep-ExtSMOTE Architecture: Dirichlet ExtSMOTE with Autoencoders . . . 150
6.2.3.1 Components of the Deep Learning Framework . . . 154
6.2.4 Principal Component Analysis (PCA) . . . 154
6.3 Simulation Results . . . 155
6.4 Application Results . . . 157
6.4.1 Scenario 1: Original Isolet Dataset . . . 158
6.4.2 Scenario 2: Reduced Isolet Dataset . . . 161
6.5 Discussion . . . 164
6.5.1 Conclusion . . . 166
Bibliography . . . 166

7 Discussion 169

7.1 Summary of Contributions and Novel Developments 170
7.1.1 Mitigate the Risks Associated with Rebalancing Imbalanced Datasets 172
7.2 Overall Impact and Future Directions 176
Bibliography 177

Appendices **179**

A **180**
A.1 Assessing Performance Across Different Dataset Shapes 181
A.2 Simulation Results for Different Accuracy Measures 183
A.3 Sensitivity Analysis of the Parameters 187
A.3.1 Parameters of Dirichlet ExtSMOTE 187
A.3.1.1 Impact of Different Multipliers (m) 187
A.3.1.2 Impact of Different Types 192
A.3.2 Parameters of FCRP SMOTE 193
A.3.2.1 Impact of Different α Values 193
A.3.3 Parameters of BGMM SMOTE 194
A.3.3.1 Impact of Different Priors 194
A.4 PCA Plots of Outliers Detected by LOF 195
A.5 Comprehensive Evaluation with Additional Accuracy Measures . . . 199
A.5.1 MCC 200
A.5.2 PR-AUC 203
Bibliography 206

List of Figures

1.1 Class imbalance problem 2
1.2 Abnormal instances in the minority class 4
1.3 Examples of class imbalance issues in various fields. 6

3.1 Word cloud of the list of 73 words after stemming. 49
3.2 Data collection and pre-processing process. 50
3.3 Time series plot for originally obtained data and the data considered for the study. 51
3.5 First five bi-words selection from each method. 57
3.6 Word cloud of the list of 44 identified symptoms. 57
3.7 Relative frequency of symptoms in long COVID patients. 58
3.8 Most frequent 15 long COVID symptoms by time. 59
3.9 Association rules visualization. 60

4.1 Identifying confirmed LCS group. LCS - Long COVID Syndrome, EMRs - Electronic Medical Records. 73
4.2 Process map 75
4.3 Agreement between nine models 83

5.1 Abnormal instances in the minority class 99
5.2 Re-sampled data with SMOTE 101
5.3 Data generation mechanisms 102
5.4 Distance ExtSMOTE data generation 104
5.5 Dirichlet ExtSMOTE (Inverse Distance) 108
5.6 FCRP SMOTE 111
5.7 BGMM SMOTE 115
5.8 Results comparison of one simulated dataset with two features and five abnormal instances 117

5.9 F1 Scores across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. 119
5.10 PR-AUCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. 120
5.11 MCCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. 121
5.12 F1 Score Ranks for the datasets with 100 × 5-fold cross validation . 125

6.1 Train the Autoencoder 150
6.2 Extract encoded representation 152
6.3 Resampling and Classification 153
6.4 F1-Score distribution for 100 trials using simulated datasets with 1000 samples and 500 features (all informative), with an imbalance ratio (IR) of 3. 156
6.5 F1-Score distribution for 100 trials using simulated datasets with 1000 samples and 5000 features (2000 informative), with an imbalance ratio (IR) of 3. 157
6.6 Scatter plot of the first two principal components in the original Isolet dataset, with the minority class in blue and detected outliers in red. 158
6.7 F1 Scores for the Isolet dataset across 50 training and test splits. . 159
6.8 Scatter plot of the first two principal components in the reduced Isolet dataset, with the minority class in blue and detected outliers in red. 161
6.9 F1 Scores for the reduced Isolet dataset across 100 training and test splits. 162

A.1 Results comparison with noisy moons dataset with one abnormal instance when noise = 0.2 181
A.2 Results comparison with noisy circles dataset with one abnormal instance when noise = 0.2. 182
A.3 F1 Scores across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. 184
A.4 PR-AUCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. 185
A.5 MCCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. 186
A.6 Mean F1 Scores for different multipliers of Dirichlet ExtSMOTE for an outlier ratio of 0.05 and two different imbalance ratios for each type. 190
A.7 Mean F1 Scores for different multipliers of Dirichlet ExtSMOTE for an outlier ratio of 0.1 and two different imbalance ratios for each type. 191
A.8 Mean F1 Scores for different types of Dirichlet ExtSMOTE for two different outlier ratios and two different imbalance ratios. 192
A.9 Mean F1 Scores for different alpha values of FCRP SMOTE for two different outlier ratios and two different imbalance ratios. 193
A.10 Mean F1 Scores for different priors of BGMM SMOTE for two different outlier ratios and two different imbalance ratios. 194
A.11 Principal component plots for the first two components, showing the distribution of outliers within the minority class across 4 datasets detected by the LOF method. 195

A.12 Principal component plots for the first two components, showing the distribution of outliers within the minority class across 8 datasets detected by the LOF method. 196
A.13 Principal component plots for the first two components, showing the distribution of outliers within the minority class across 8 datasets detected by the LOF method. 197
A.14 Principal component plots for the first two components, showing the distribution of outliers within the minority class across 6 datasets detected by the LOF method. 198

List of Tables

2.1 Categorization of SMOTE algorithm extensions 24
2.2 Confusion Matrix 34

3.1 The list of words explains users' opinions rather than the experience. 46
3.2 List of long COVID symptoms identified by different sources in the literature. 47
3.3 Pre-processed word corpus of stemmed symptoms. 48
3.4 Bigram contingency table for a bigram (xy): observed frequencies (O_{ij}). z_* indicates words that are not x nor y. [a] 54
3.5 Bigram contingency table for a bigram (xy): estimated frequencies (E_{ij}). [b] 55
3.6 List of identified symptom rules. 60

4.1 MaPCReN patients with encounter note records in April-June 2022 72
4.2 Identified LCS patient counts and percentages with model accuracy measures 82
4.3 Demographic information of each patient group 83
4.4 Pre-COVID symptom classification of each patient group 84
4.5 Post-COVID symptom classification of each patient group 85
4.6 Overall classification of symptoms with percentage difference 86

5.1 Characteristics of the binary class datasets used in the computational study. 123
5.2 Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on Logistic classifier. . . . 126
5.3 Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on KNN classifier. 127
5.4 Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on Random Forest classifier. 127

5.5 F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Logistic Regression classifier. 129
5.6 F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a K-NN classifier. . 130
5.7 F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier. 131

6.1 Comparison of resampling and feature reduction methods for the original Isolet data. Best techniques and methods are highlighted, with Deep-ExtSMOTE in red and statistical significance in the last column. 160
6.2 Comparison of resampling and feature reduction methods for the reduced Isolet data. Best techniques and methods are highlighted, with Deep-ExtSMOTE in red and statistical significance in the last column. 163

A.1 MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Logistic Regression classifier. 200
A.2 MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a K-NN classifier. 201
A.3 MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier. 202
A.4 PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Logistic Regression classifier. 203
A.5 PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a K-NN classifier. . 204
A.6 PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier. 205

Chapter 1

Introduction

1.1 Class Imbalance

In recent years, the rapid advancement of science and technology has resulted in the generation of increasingly complex datasets. These datasets, rich in diversity and scope, hold immense potential for driving innovation and uncovering valuable insights across various fields. One of the potential areas of exploration in leveraging these complex datasets is predictive modeling, which focuses on predicting outcomes and behaviors based on existing data, thereby enhancing our ability to make data-driven decisions and solve complex problems. However, these opportunities also bring significant challenges.

One of the most critical challenges is the class imbalance problem (Niaz et al.; 2022). This issue arises when the distribution of data across different classes or categories is uneven, often resulting in a scenario where one class significantly outnumbers the others. In binary classification, the class with the most instances is referred to as the 'majority class', while the class with fewer instances is known as the 'minority class'. This imbalance can skew the performance of machine learning models, causing them to favor the majority class and overlook or misclassify the minority class (Wu and Chang; 2003). Fig. 1.1 illustrates this issue with a scatter plot of two features, clearly showing the disparity between the minority and majority classes.

When dealing with only two classes, this issue is known as a binary class imbalance problem (Fernández et al.; 2011; Chawla et al.; 2004). Conversely, when more than two classes are involved, it is called a multi-class imbalance problem

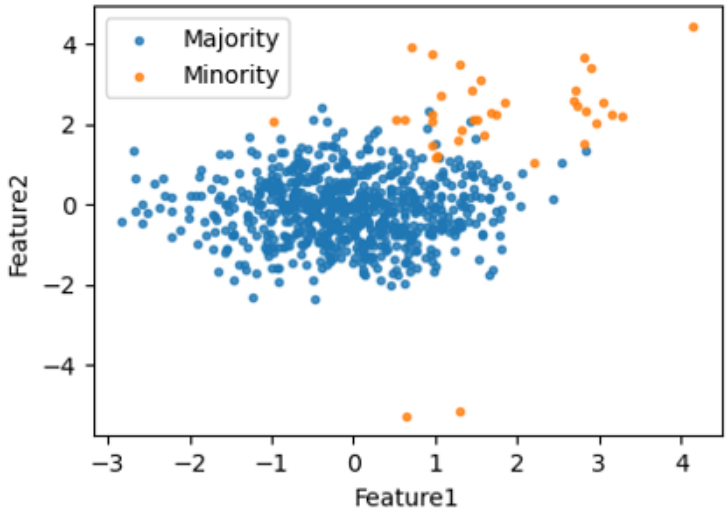


Figure 1.1: Class imbalance problem

(Wang and Yao; 2012; Rachmatullah; 2022). Addressing this issue is crucial, as learning from imbalanced datasets is recognized as one of the top ten challenging problems in data mining research (Yang and Wu; 2006).

Additional factors, such as outliers (Nnamoko and Korkontzelos; 2020; Turlapati and Prusty; 2020) and the "curse of dimensionality" (Bellman; 1957), compound the class imbalance problem. Outliers, which are data points that deviate significantly from other observations (Hawkins; 1980), can skew results and affect model performance. Meanwhile, the curse of dimensionality refers to the challenges associated with high-dimensional data, where the number of features or attributes is large relative to the number of observations. This can complicate the identification of patterns and relationships in the data. Additionally, in high-dimensional spaces, models are prone to overfitting, where they capture noise rather than the true underlying patterns. As the dimensionality increases, the model's complexity grows, making it more likely to overfit the training data, which can negatively impact the model's generalization to new, unseen data.

Addressing the challenges in class imbalance data is crucial for developing robust and accurate classification models. Effective solutions must account for the imbalance in data distribution, mitigate the impact of outliers, and handle high-dimensional data effectively. By tackling these issues, we strive to enhance model performance and derive more reliable insights from imbalanced datasets. This thesis is dedicated to exploring and developing new methods to effectively address

these challenges.

1.1.1 Challenges in Handling Imbalanced Data

Imbalanced data presents significant challenges for predictive modeling, especially when one class is significantly underrepresented. One major issue is that standard metrics like accuracy can be misleading (Chawla et al.; 2002; Japkowicz and Stephen; 2002). A model might achieve high accuracy simply by predicting the majority class, failing to reflect its performance on the minority class. To address this, it is crucial to use metrics such as precision, recall, F1-score, and the Matthew's Correlation Coefficient for a more balanced evaluation, thereby overcoming the limitations of standard metrics like accuracy. We discuss these performance measures in detail in Section 2.5.

Imbalanced datasets have a profound impact on the performance of classifiers (Luque et al.; 2019). Models trained on such data often struggle with poor performance in the minority class due to bias towards the majority class. This results in higher false negative rates and lower true positive rates for the minority class, which is critical in fields like disease diagnosis and fraud detection (Awoyemi et al.; 2017; Yang et al.; 2009). Additionally, the limited number of minority class instances can lead to overfitting and challenge the effectiveness of certain algorithms. Techniques like resampling (oversampling or undersampling) and synthetic data generation (e.g., SMOTE) are often used to address these issues.

The bias in imbalanced datasets can skew decision boundaries, leading to suboptimal predictions and difficulty distinguishing between classes. Class overlap, where features of the minority and majority classes are similar, further complicates accurate prediction. Real-world constraints often make it impractical to acquire more minority class data, highlighting the need for robust algorithms and techniques that can handle imbalance without additional data. This practical approach is a key focus of our research. Effective evaluation of predictive models requires careful cross-validation and appropriate metrics to ensure that both classes are well-represented and assessed. Addressing these challenges involves combining advanced sampling techniques, robust metrics, and tailored algorithmic adjustments to improve model accuracy and decision-making across various domains.

1.2 Abnormal Instances

In binary classification, abnormal instances refer to data points that deviate significantly from most of the dataset or are incorrectly labelled (Fig. 1.2). These instances can be outliers, noise, or mislabeled data, causing challenges in building accurate predictive models.

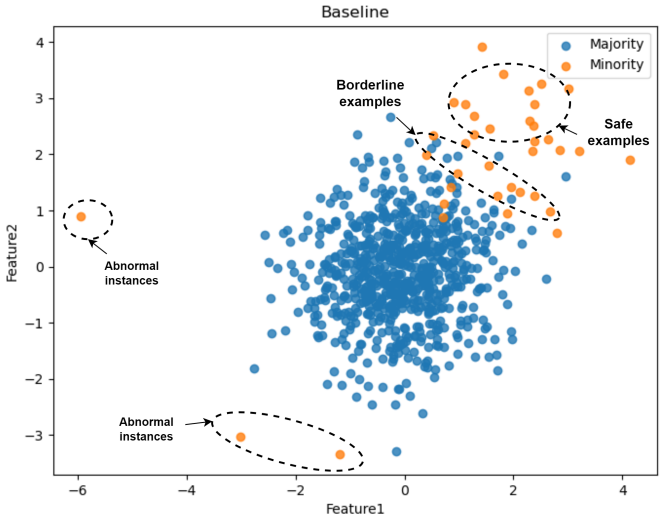


Figure 1.2: Abnormal instances in the minority class

Outliers can represent legitimate extreme observations resulting from random fluctuations and thus reflect the inherent characteristics of a random sampling process. In such cases, retaining these outliers and treating them on par with all other data points in the analysis is recommended (Napierala and Stefanowski; 2016; Kubat and Matwin; 1997). Noise is an aspect of data points, which means these points don't follow the general pattern of the dataset. Mislabelled data refers to instances in a dataset that have been incorrectly labelled or assigned a wrong class or category during data collection.

In a medical diagnosis task for identifying whether a patient has a specific disease or not based on various health parameters, an abnormal instance could be a data point that contradicts the common characteristics associated with the disease. For instance, if most patients diagnosed with a certain illness exhibit specific symptoms like fever, fatigue, and cough, an abnormal instance might be a patient

labelled as positive for the disease but who lacks these typical symptoms or shows contradictory symptoms not usually associated with the illness. Such instances might distort the statistical properties of the dataset, affecting the distribution of this particular feature. If not handled appropriately, these abnormal instances could mislead the learning algorithm and bias the model towards making incorrect predictions. We can also refer to studies about medical problems by Gamberger et al. (1999), where the authors showed that the results of noise identification filters were often confirmed by experts as valid outliers.

Abnormal instances, by their very nature, reside at the outer edges of the data distribution and often represent unique and anomalous instances within the minority class. When SMOTE, a popular synthetic data generation tool for handling imbalance (Chawla et al.; 2002) is applied to datasets containing such data points, it can inadvertently generate synthetic samples that fail to capture the genuine characteristics of the minority class due to the influence of these abnormal instances. As a result, any subsequent machine learning model's ability to make accurate predictions on minority class instances is compromised.

To ensure the reliable application of SMOTE in practical scenarios, it is crucial to develop effective strategies for tackling the challenge posed by abnormal instances within the minority class. In addressing this issue, we consider the significance of each instance, assigning appropriate weights to neighbouring points to mitigate this problem effectively.

1.3 Motivating Examples

In numerous fields, the unequal distribution of classes within datasets poses significant challenges to accurate model prediction. This disparity can lead to biased outcomes, particularly when the minority class is of critical importance, as seen in medical diagnosis and fraud detection. Furthermore, real-world data often includes abnormal instances within the minority class, a natural occurrence that complicates the classification task even further (Napierala and Stefanowski; 2016). Additionally, datasets can be high-dimensional, adding another layer of complexity to the problem. The Fig. 1.3 illustrates examples of class imbalance across various domains, and later sections explore specific cases in finance, biomedicine, security

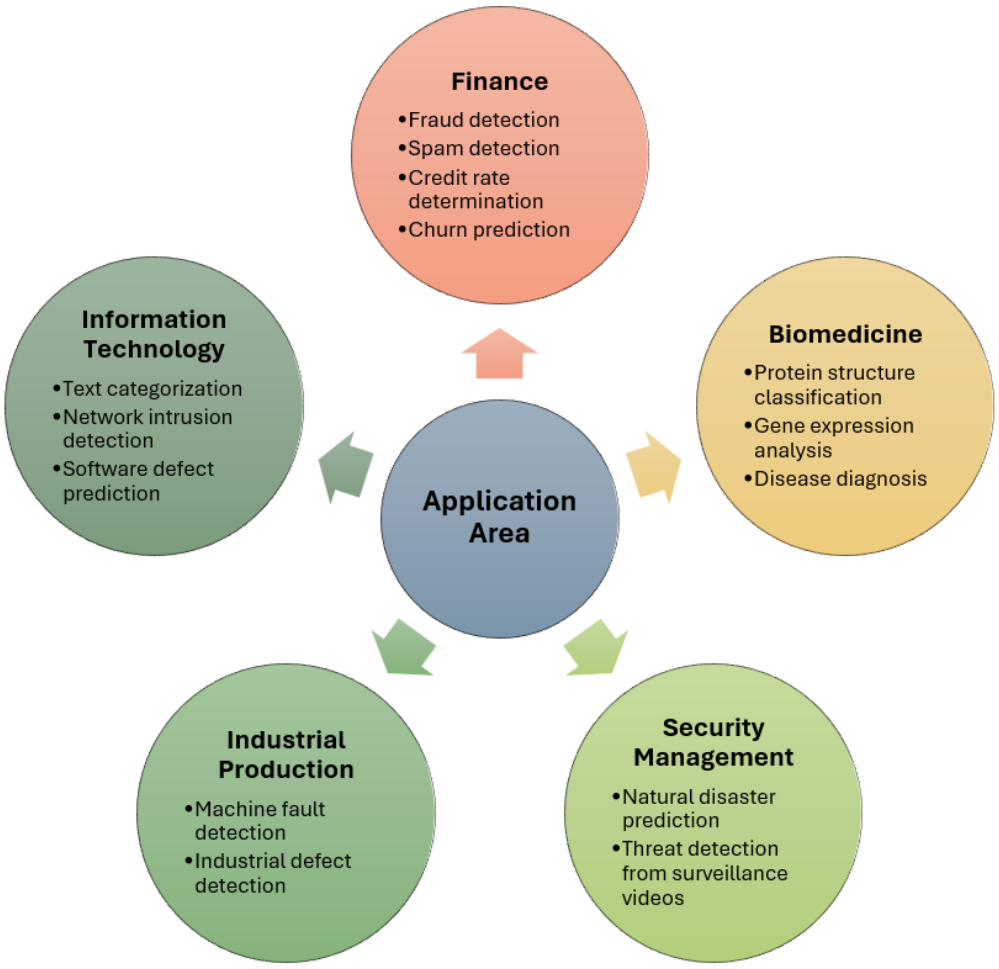


Figure 1.3: Examples of class imbalance issues in various fields.

management, industrial production, and information technology, underscoring the critical need for effective solutions.

Finance

In the finance sector, class imbalance is a notable challenge across various applications (He and Garcia; 2009). Fraud detection (Awoyemi et al.; 2017; Dal Pozzolo et al.; 2014; Yang et al.; 2009) is a prime example, where the vast majority of transactions are legitimate, making fraudulent transactions rare. This imbalance makes it difficult for predictive models to accurately identify fraudulent activities, potentially leading to significant financial losses if not properly addressed. Spam

detection (Liu et al.; 2017) also faces similar issues, with the majority of emails being legitimate and only a small proportion being spam. Effective spam filtering is crucial to maintain efficient communication and protect users from phishing and other malicious activities. In credit rate determination (Zhang and Liu; 2019), the minority class of defaulted loans is often overshadowed by the majority of non-defaulted ones, complicating the accurate assessment of creditworthiness. Finally, churn prediction (Zhu et al.; 2018) involves identifying customers likely to leave a service, which is challenging due to the relatively small number of churn instances compared to the large number of loyal customers. However, customer retention is important for maintaining business, as it is easier to retain customers than recruit new ones. Addressing class imbalance in these areas is essential for improving fraud detection, spam filtering, credit scoring, and customer retention.

Furthermore, finance examples can effectively highlight the challenges posed by outliers in the minority class and the complexities of high-dimensional data. For instance, in fraud detection, fraudulent transactions represent the minority class and often exhibit unusual patterns, such as sudden large transfers or atypical purchases, which are outliers compared to normal behavior. These outliers, if not properly managed, can lead to incorrect classifications and compromise the detection system (Padmaja et al.; 2007; Baesens et al.; 2021). Similarly, financial datasets are inherently high-dimensional, involving numerous variables like transaction histories, account activities, and demographic details. This high dimensionality makes it difficult to detect meaningful patterns, especially when outliers disrupt the data's structure (Zhou et al.; 2017; Ando; 2016). In credit scoring, for example, an outlier like an unexpected spike in credit usage can skew the analysis, leading to inaccurate assessments of an individual's creditworthiness. (Qian et al.; 2021). These finance examples underscore the critical need for robust methods to handle outliers and high-dimensional data to ensure accurate and reliable financial decision-making.

Biomedicine

Class imbalance presents significant challenges in several biomedical applications (Rahman and Davis; 2013). In protein structure classification (Zhao et al.; 2008), rare protein structures are often underrepresented in datasets, making it difficult to classify them accurately and hindering advancements in understanding protein functions and drug design. Gene expression analysis (Reyes-Nava et al.;

2019) can also suffer from imbalances, especially when certain gene expressions are rare but critical for understanding diseases. This imbalance can lead to poor predictive model performance, impacting research and treatment development. Disease diagnosis (Yu et al.; 2013; Khaldy and Kambhampati; 2018; Mazurowski et al.; 2008; Matharaarachchi, Domaratzki, Katz and Muthukumarana; 2024) is another area where class imbalance is prevalent, particularly with rare diseases or conditions. For example, cancer detection involves identifying a small number of positive cases among a large number of healthy instances, complicating accurate diagnosis. Addressing class imbalance in these contexts is crucial for developing reliable diagnostic tools, advancing research, and improving patient outcomes through accurate detection and treatment planning.

Biomedicine also highlights the challenges posed by outliers in the minority class and high-dimensional data. In the study by Ijaz et al. (2020) the outliers discussed are likely data points that do not conform to the expected patterns within the dataset used for cervical cancer prediction. These outliers could include unusual or extreme cases that deviate significantly from the typical characteristics of patients with or without cervical cancer. For example, outliers might be instances where patients exhibit atypical combinations of risk factors, symptoms, or test results that differ from the majority of cases. Moreover, biomedical datasets, particularly in genomics, are high-dimensional, containing thousands of genetic markers per patient (Yu and Ni; 2014). These outliers, along with the vast number of features, complicate analysis, emphasizing the need for robust methods to ensure accurate diagnostics and effective treatment strategies.

Security Management

In security management, class imbalance affects several critical areas. Natural disaster prediction (Eltehewy et al.; 2023) relies on predicting rare events from vast amounts of meteorological data. Accurate prediction is essential for effective disaster management and mitigation strategies. Similarly, threat detection from surveillance videos (Choudhry et al.; 2023) involves analyzing large volumes of benign footage with only a tiny fraction indicating potential threats. Effective detection of these rare malicious activities is crucial for maintaining robust security systems. Solutions to the class imbalance in security management can greatly enhance the effectiveness of security measures, ensuring timely responses to natural disasters and maintaining safety through reliable threat detection.

Security management provides clear examples of the challenges posed by outliers in the minority class and high-dimensional data. In areas like intrusion detection (Jin et al.; 2021), rare but critical events, such as cyberattacks, represent outliers within the minority class. These events deviate from normal patterns and are difficult to detect due to their rarity. Additionally, security data is often high-dimensional (Cui et al.; 2023), involving numerous features like user behavior logs and network traffic data, which complicates the identification of these outliers. Addressing these challenges is essential for maintaining robust security systems.

Information Technology

In information technology, class imbalance is evident in several applications. Text categorization (Chen et al.; 2011) often deals with common categories vastly outnumbering rare ones, complicating accurate categorization efforts. Network intrusion detection (Yuan et al.; 2017) faces the challenge of identifying rare malicious activities among the majority of benign network activities. Accurate detection of these rare threats is crucial for maintaining network security. Additionally, software defect prediction (Chen et al.; 2018; Wang and Yao; 2013; Feng et al.; 2021) involves identifying rare defects within a large amount of defect-free code. Addressing class imbalance in these IT applications improves predictive model performance, enhancing text categorization, network security, and software reliability.

IT examples in network security, system monitoring, and anomaly detection highlight the challenges of outliers in the minority class and high-dimensional data. These outliers, such as system failures, unusual user behavior, or cyberattacks, deviate from normal patterns and are crucial to detect for maintaining security and stability. The high-dimensional nature of IT data, including logs and user activity, further complicates anomaly detection due to the high number of features, many of which are irrelevant to predicting defects (Bahaweres et al.; 2021). To address this, Tang and Khoshgoftaar (2004) proposed treating noise samples as outliers, using a noise factor (NF) to identify and remove the most significant outliers.

Addressing class imbalance is not only essential for improving predictive model performance but also for enhancing decision-making and outcomes across diverse applications. By developing and implementing robust techniques that account for the presence of abnormal instances in the minority class and the challenges

of high-dimensional data, we can achieve more accurate and reliable predictions, ultimately benefiting society as a whole.

1.3.1 Identifying Long COVID Patients: Challenges of Imbalanced Data

Identifying long COVID patients presents a complex challenge, primarily due to the significant imbalance in the dataset, where identified long COVID cases are vastly outnumbered by non-long COVID cases (Matharaarachchi, Domaratzki, Katz and Muthukumarana; 2024). Traditional classification algorithms often struggle with such imbalanced data, as they tend to focus on the majority class, resulting in biased outcomes that do not accurately represent the minority class. This bias leads to the frequent misclassification of long COVID patients, whose unique and diverse symptoms are often overshadowed by the more prevalent non-long COVID cases. The variability and lack of specificity of long COVID symptoms further compound this issue, making it difficult for algorithms to distinguish these patients accurately. As a result, the nuanced patterns specific to long COVID are often missed, exacerbating the misclassification problem.

Moreover, the challenge of imbalanced data in identifying long COVID patients is not merely a technical issue but also has significant clinical implications. The failure to correctly identify long COVID cases can lead to a lack of appropriate medical attention and support for these patients, potentially worsening their health outcomes and quality of life. Therefore, there is a pressing need for specific approaches that can effectively handle data imbalance. This thesis considers finding solutions to this real-time problem, emphasizing the improvement of minority class detection without compromising overall accuracy. By ensuring that the unique characteristics of long COVID are recognized and appropriately managed, these methods contribute to better healthcare outcomes and a deeper understanding of this condition. Overcoming the challenges of imbalanced data is thus crucial for the accurate identification and treatment of long COVID patients.

1.4 Objective of the Thesis

The primary objectives of this research are as follows:

- To provide insights into the challenges of abnormal instances within the minority class when using SMOTE and some popular existing extensions.
- To propose innovative methods, algorithms and strategies capable of mitigating the adverse effects of abnormal instances on class imbalance data.
- To offer empirical evidence, supported by simulated and experimental results, that demonstrates the effectiveness of these proposed solutions in enhancing classification performance.
- To design and implement methods that integrate autoencoder-based feature reduction with extended versions of proposed techniques to enhance reliability in high-dimensional contexts.

1.5 Main Contributions and Thesis Organization

In Chapter 1 and Chapter 2, we introduce the problem and discuss the related background methodologies to set the stage for our research into addressing class imbalance in machine learning. These chapters lay out foundational concepts and previous work in resampling techniques. In this thesis, our main contributions focus on proposing various resampling techniques to tackle imbalance, especially in the presence of abnormal instances in the minority class and with high-dimensional data. Each proposed technique achieves this through a distinct strategy. The following is a summary of our contributions, presented in Chapters 3 to 7.

Chapter 3: Discovering Long COVID Symptom Patterns

Chapter 3 explores the background literature necessary for identifying long COVID patients in Manitoba, as addressed in Chapter 4. This exploration was crucial because, at the time, long COVID symptoms were not well-defined in the available literature.

Building on the published paper "Discovering Long COVID Symptom Patterns: Association Rule Mining and Sentiment Analysis in Social Media Tweets" (Matharaarachchi et al.; 2022), Chapter 3 delves into Long COVID Syndrome (LCS) through the analysis of symptoms reported on Twitter over a year. Using natural language processing (NLP) and association rule mining techniques (Agrawal

et al.; 1993; Celi et al.; 2020; Keselj and Martin; 2009), this study aims to uncover patterns and behaviors associated with LCS. The findings contribute to a better understanding of the condition, potentially aiding in early detection and personalized treatment strategies.

Chapter 4: Long COVID Prediction in Manitoba

Chapter 4 discusses the paper in review, "Long COVID Prediction in Manitoba Using Clinical Notes Data: A Machine Learning Approach" (Matharaarachchi, Domaratzki, Katz and Muthukumarana; 2024). Building on the insights from Chapter 3, this study focuses on predicting long COVID cases using machine learning classification models applied to clinical notes data. A significant challenge addressed in this work is the class imbalance within the dataset, where long COVID cases are underrepresented. To mitigate this, existing techniques are used to prevent model bias toward the majority class, ensuring accurate identification of minority class instances. Despite the challenges of imbalance, the study achieves high predictive accuracy by integrating patient attributes and symptoms associated with long COVID.

Chapter 5: Enhancing SMOTE for Imbalanced Data with Abnormal Minority Instances

After exploring two-class classification in the context of long COVID in Chapter 4, we now turn our attention to more advanced problems, focusing on developing novel methods to further address class imbalance issue in classification, particularly when abnormal instances appear in the minority class.

Chapter 5 presents the paper in review, "Enhancing SMOTE for Imbalanced Data with Abnormal Minority Instances" (Matharaarachchi, Domaratzki and Muthukumarana; 2024). An editor requested revision of this paper has been submitted. This chapter introduces innovative extensions of the Synthetic Minority Oversampling Technique (SMOTE) to enhance its performance in the presence of abnormal instances in minority class. The proposed extensions—Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE, and BGMM SMOTE—demonstrate superior classification performance compared to SMOTE and its extensions through comprehensive experiments on simulated and real-world datasets.

Chapter 6: Deep-ExtSMOTE: Integrating Autoencoders for Advanced Mitigation of Class Imbalance in High-Dimensional Data Classification

Chapter 6 consists of the to-be-submitted work "Deep-ExtSMOTE: Integrating Autoencoders for Advanced Mitigation of Class Imbalance in High-Dimensional Data Classification." This research aims to address the challenges posed by class imbalance in high-dimensional classification tasks by combining autoencoder-based feature reduction with a proposed extended version of SMOTE. The goal is to enhance the generation of synthetic samples and improve the reliability of machine learning models.

Chapter 7: Discussion

Finally, in Chapter 7, we summarize the thesis with concluding remarks and discuss future research directions. We reflect on the methodological contributions presented in Chapters 3 to 6 and their implications for addressing challenges in imbalanced datasets with abnormal instances, and high-dimensional features. We highlight areas for further investigation and development, aiming to advance the field of machine learning and data science.

In conclusion, this thesis provides a thorough exploration of innovative methodologies to tackle the intertwined challenges of class imbalance, abnormal instances, and high-dimensional data in machine learning. The novel approaches introduced here significantly advance the field, contributing to the development of more robust and reliable machine learning models for real-world applications.

1.6 Contributions of Authors

This sandwich thesis was made possible through the significant contributions of each author. Surani Matharaarachchi played a leading role in this research, including conceptualization, methodology, execution of the research investigation, implementation, programming, formal analysis, data curation, software development, validation, and visualization. Surani also drafted the initial manuscripts and the entire thesis, and managed the integration of feedback from all authors, ensuring coherence and consistency across the research papers.

Dr. Saman Muthukumarana and Dr. Mike Domaratzki were instrumental in conceptualizing and supervising the research, providing methodological input, and developing the initial research framework and theoretical foundations. Dr. Alan Katz's expert contributions were crucial in obtaining data, formulating research questions, and interpreting the results, particularly in Chapters 3 and 4. Their guidance was essential in shaping the research framework and establishing the theoretical foundations. The collaborative efforts of all contributors were instrumental in achieving the research objectives and producing a comprehensive and cohesive thesis.

Bibliography

Agrawal, R., Imieliński, T. and Swami, A. (1993). Mining association rules between sets of items in large databases, *SIGMOD record* **22**(2): 207–216.

Ando, S. (2016). Classifying imbalanced data in distance-based feature space, *Knowledge and information systems* **46**(3): 707–730.

Awoyemi, J. O., Adetunmbi, A. O. and Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis, *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pp. 1–9.

Baesens, B., Höppner, S., Ortner, I. and Verdonck, T. (2021). robrose: A robust approach for dealing with imbalanced data in fraud detection, *Statistical methods applications* **30**(3): 841–861.

Bahaweres, R. B., Jana, E. D. H., Hermadi, I., Suroso, A. I. and Arkeman, Y. (2021). Handling high-dimensionality on software defect prediction with flda, *2021 2nd International Conference On Smart Cities, Automation Intelligent Computing Systems (ICON-SONICS)*, pp. 76–81.

Bellman, R. (1957). *Dynamic programming.*, Princeton University Press, Princeton.

Celi, L. A., Majumder, M. S., Ordóñez, P., Osorio, J. S., Paik, K. E. and Somai, M. (2020). Introduction to clinical natural language processing with python,

Leveraging Data Science for Global Health, Springer International Publishing AG, Switzerland.

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

Chawla, N. V., Japkowicz, N. and Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets, *SIGKDD explorations* **6**(1): 1–6.

Chen, E., Lin, Y., Xiong, H., Luo, Q. and Ma, H. (2011). Exploiting probabilistic topic models to improve text categorization under class imbalance, *Information Processing Management* **47**(2): 202–214.

Chen, L., Fang, B., Shang, Z. and Tang, Y. (2018). Tackling class overlap and imbalance problems in software defect prediction, *Software quality journal* **26**(1): 97–125.

Choudhry, N., Abawajy, J., Huda, S. and Rao, I. (2023). A comprehensive survey of machine learning methods for surveillance videos anomaly detection, *IEEE access* **11**: 114680–114713.

Cui, J., Zong, L., Xie, J. and Tang, M. (2023). A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data, *Applied intelligence (Dordrecht, Netherlands)* **53**(1): 272–288.

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S. and Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective, *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD 2014)*, IEEE, pp. 1–10.

Eltehewy, R., Abouelfarag, A. and Saleh, S. N. (2023). Efficient classification of imbalanced natural disasters data using generative adversarial networks for data augmentation, *ISPRS international journal of geo-information* **12**(6): 245–.

Feng, S., Keung, J., Yu, X., Xiao, Y., Bennin, K. E., Kabir, M. A. and Zhang, M. (2021). Coste: Complexity-based oversampling technique to alleviate the class imbalance problem in software defect prediction, *Information and Software Technology* **129**: 106432.
URL: *https://www.sciencedirect.com/science/article/pii/S0950584920301889*

Fernández, A., García, S. and Herrera, F. (2011). Addressing the classification with imbalanced data: Open problems and new challenges on class distribution, *Hybrid Artificial Intelligent Systems*, Vol. 6678 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–10.

Gamberger, D., Boskovic, R., Lavrač, N. and Groselj, C. (1999). Experiments with noise filtering in a medical domain, *Proceedings of the 16th International Conference on Machine Learning (ICML)*, Morgan Kaufmann, pp. 143–151.

Hawkins, D. (1980). *Identification of Outliers*, Monographs on Applied Probability and Statistics, Springer Netherlands, Dordrecht.

He, H. and Garcia, E. (2009). Learning from imbalanced data, *IEEE transactions on knowledge and data engineering* **21**(9): 1263–1284.

Ijaz, M. F., Attique, M. and Son, Y. (2020). Data-driven cervical cancer prediction model with outlier detection and over-sampling methods, *Sensors (Basel, Switzerland)* **20**(10): 2809–.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study, *Intelligent data analysis* **6**(5): 429–449.

Jin, F., Chen, M., Zhang, W., Yuan, Y. and Wang, S. (2021). Intrusion detection on internet of vehicles via combining log-ratio oversampling, outlier detection and metric learning, *Information Sciences* **579**: 814–831.
URL: *https://www.sciencedirect.com/science/article/pii/S0020025521008021*

Keselj, Vlado (Review of: Jurafsky, D. and Martin, J. H. (2009). Speech and language processing (second edition), *Computational Linguistics* **35**(3): 463–466.

Khaldy, M. A. I. and Kambhampati, C. (2018). Resampling imbalanced class and the effectiveness of feature selection methods for heart failure dataset, *International Robotics Automation Journal* **4**(1): 37–45.

Kubat, M. and Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection, *Proceedings of the 14th International Conference on Machine Learning*, Morgan Kaufmann, pp. 179–186.

Liu, S., Wang, Y., Zhang, J., Chen, C. and Xiang, Y. (2017). Addressing the class imbalance problem in twitter spam detection using ensemble learning, *Computers security* **69**: 35–49.

Luque, A., Carrasco, A., Martín, A. and de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix, *Pattern recognition* **91**: 216–231.

Matharaarachchi, S., Domaratzki, M., Katz, A. and Muthukumarana, S. (2022). Discovering long covid symptom patterns: Association rule mining and sentiment analysis in social media tweets, *JMIR formative research* **6**(9): e37984–e37984.

Matharaarachchi, S., Domaratzki, M., Katz, A. and Muthukumarana, S. (2024). Long covid prediction in manitoba using clinical notes data: A machine learning approach (in review).

Matharaarachchi, S., Domaratzki, M. and Muthukumarana, S. (2024). Enhancing smote for imbalanced data with abnormal minority instances, *Machine learning with applications* **18**: 100597–.

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A. and Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance, *Neural networks : the official journal of the International Neural Network Society* **21**(2-3): 427—436.

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.

Niaz, N. U., Shahariar, K. N. and Patwary, M. J. A. (2022). Class imbalance problems in machine learning: A review of methods and future challenges, *Proceedings of the 2nd International Conference on Computing Advancements*, ACM, New York, NY, USA, pp. 485–490.

Nnamoko, N. and Korkontzelos, I. (2020). Efficient treatment of outliers and class imbalance for diabetes prediction, *Artificial intelligence in medicine* **104**: 101815–101815.

Padmaja, T. M., Dhulipalla, N., Bapi, R. S. and Krishna, P. R. (2007). Unbalanced data classification using extreme outlier elimination and sampling techniques

for fraud detection, *15th International Conference on Advanced Computing and Communications (ADCOM 2007)*, pp. 511–516.

Qian, H., Zhang, S., Wang, B., Peng, L., Gao, S. and Song, Y. (2021). A comparative study on machine learning models combining with outlier detection and balanced sampling methods for credit scoring, *arXiv.org* .

Rachmatullah, M. (2022). The application of repeated smote for multi class classification on imbalanced data, *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer* **22**.

Rahman, M. M. and Davis, D. N. (2013). Addressing the class imbalance problem in medical datasets, *International Journal of Machine Learning and Computing* pp. 224–228.

Reyes-Nava, A., Cruz-Reyes, H., Alejo, R., Rendón-Lara, E., Flores-Fuentes, A. A. and Granda-Gutiérrez, E. E. (2019). Using deep learning to classify class imbalanced gene-expression microarrays datasets, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 46–54.

Tang, W. and Khoshgoftaar, T. (2004). Noise identification with the k-means algorithm, *16th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, Los Alamitos CA, pp. 373–378.

Turlapati, V. P. K. and Prusty, M. R. (2020). Outlier-smote: A refined oversampling technique for improved detection of covid-19, *Intelligence-based medicine* **3-4**: 100023–100023.

Wang, S. and Yao, X. (2012). Multiclass imbalance problems: Analysis and potential solutions, *IEEE transactions on systems, man and cybernetics. Part B, Cybernetics* **42**(4): 1119–1130.

Wang, S. and Yao, X. (2013). Using class imbalance learning for software defect prediction, *IEEE Transactions on Reliability* **62**(2): 434–443.

Wu, G. and Chang, E. (2003). Class-boundary alignment for imbalanced dataset learning.

Yang, Q. and Wu, X. (2006). 10 challenging problems in data mining research, *International Journal of Information Technology Decision Making (IJITDM)* **05**(04): 597–604.

Yang, Z., Tang, W., Shintemirov, A. and Wu, Q. (2009). Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **39**: 597 – 610.

Yu, H. and Ni, J. (2014). An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **11**(4): 657–666.

Yu, H., Ni, J. and Zhao, J. (2013). Acosampling: An ant colony optimization-based undersampling method for classifying imbalanced dna microarray data, *Neurocomputing (Amsterdam)* **101**: 309–318.

Yuan, X., Li, C. and Li, X. (2017). Deepdefense: Identifying ddos attack via deep learning, *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1–8.

Zhang, H. and Liu, Q. (2019). Online learning method for drift and imbalance problem in client credit assessment, *Symmetry (Basel)* **11**(7): 890–.

Zhao, X.-M., Li, X., Chen, L. and Aihara, K. (2008). Protein classification with imbalanced data, *Proteins, structure, function, and bioinformatics* **70**(4): 1125–1132.

Zhou, P., Hu, X., Li, P. and Wu, X. (2017). Online feature selection for high-dimensional class-imbalanced data, *Knowledge-based systems* **136**: 187–199.

Zhu, B., Baesens, B., Backiel, A. and vanden Broucke, S. K. L. M. (2018). Benchmarking sampling techniques for imbalance learning in churn prediction, *Journal of the Operational Research Society* **69**(1): 49–65.

Chapter 2

Background

This chapter provides the essential background relevant to the work presented in Chapters 3 through 6. It covers key areas, including current approaches to handling data imbalance, the outlier identification techniques used for visualization, the dimensionality reduction methods used in Chapter 6, the important classifiers integrated throughout the thesis, and the performance evaluation metrics used to assess model effectiveness.

2.1 Current Approaches

In imbalanced datasets, a key challenge arises from standard classifiers potentially disregarding the significance of the minority class due to its limited representation. These classifiers often exhibit bias towards the majority class, prioritizing overall classification accuracy. As a result, instances from the minority class are frequently misclassified compared to those from the majority class. (Huang et al.; 2006).

Numerous approaches have been developed to tackle the issue of class imbalance. These methods generally fall into three main categories (Barandela et al.; 2003):

1. **Data level approaches:** Data-level approaches focus on directly modifying the dataset to address the class imbalance (Ganganwar; 2012). One common technique is resampling, which adjusts the distribution of classes within the data. Oversampling involves increasing the number of instances in the minority class, often using methods like SMOTE (Synthetic Minority

Over-sampling Technique) (Chawla et al.; 2002) to generate synthetic examples by interpolating between existing samples. For instance, in a fraud detection system where fraudulent transactions are rare, SMOTE can help create additional examples similar to other fraud instances, improving the model's ability to identify such transactions (Sisodia et al.; 2017; Ileberi et al.; 2021). Undersampling reduces the number of instances in the majority class to balance the dataset, which can be useful in cases like predicting rare diseases, where reducing the number of healthy cases ensures that the model pays more attention to the minority class (Dai and Hua; 2016; Li et al.; 2017). Data augmentation, another data-level approach, involves creating new instances through transformations such as rotation, scaling, flipping, or generating synthetic data, thereby expanding the dataset and enhancing model performance (Wong et al.; 2016; Jiang and Ge; 2021; Saini and Susan; 2020). This category of methods preprocesses the dataset by altering the data distribution to adjust the balance between classes, considering the local characteristics of individual examples (Batista et al.; 2004).

2. **Algorithmic level approaches:** Algorithm-level approaches modify the learning algorithm to better handle class imbalance without changing the dataset (Chawla et al.; 2004). One effective method is adjusting class weights, where the algorithm assigns higher importance to the minority class during training (Lee et al.; 2017). For example, in a medical diagnosis scenario where positive cases are rare, increasing the weight for positive cases can help the model learn to identify these cases more effectively. Another technique is cost-sensitive learning (Elkan; 2001), which incorporates the cost of misclassifying minority class instances into the training process. This approach ensures that the model is penalized more for incorrect predictions of minority class examples, making it more sensitive to these cases. Ensemble methods, such as bagging (Breiman; 1996) and boosting (Freund and Schapire; 1997), can also be used, where multiple models are trained to address the class imbalance. For instance, boosting algorithms like AdaBoost sequentially focus on correcting errors made by previous models, which helps improve the model's performance on minority class instances (Freund and Schapire; 1997). This category of methods aims to modify search techniques or classification decision strategies to bias towards the minority class or enhance prediction performance through weighted adjustments for each class

(Huang et al.; 2006).

3. **Hybrid approaches:** Hybrid approaches combine elements of both data-level and algorithm-level strategies to address class imbalance. For example, resampling combined with algorithmic adjustments involves using oversampling or undersampling to balance the dataset and then applying class weight adjustments to enhance model performance further (Seiffert et al.; 2010). In practice, this might mean using SMOTE to generate synthetic data and then adjusting class weights in the classifier to improve its sensitivity to the minority class (Chawla et al.; 2003). Another hybrid approach is ensemble methods with resampling (Kumaran et al.; 2024), where multiple models are trained on different balanced subsets of the data created through resampling techniques. For instance, in a fraud detection system, one might use bagging with multiple balanced datasets generated by SMOTE to improve the overall performance and robustness of the detection system. These hybrid strategies leverage the strengths of both data-level and algorithm-level methods to provide a solution to class imbalance challenges. Additionally, cost-sensitive learning solutions combine data-level and algorithmic approaches, recognizing the higher misclassification costs associated with minority class samples. They aim to mitigate errors that carry higher misclassification costs with samples in the minority class (Sun et al.; 2007).

2.1.1 SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is a widely used method to address class imbalance by generating synthetic samples for the minority class. It creates new instances along the line segments between existing minority class instances. However, SMOTE has its own limitations (Fernandez et al.; 2018):

- **Sensitivity to outliers:** SMOTE can generate synthetic samples that are not representative if outliers are present in the minority class. These outliers can lead to the creation of synthetic samples that do not generalize well to the actual minority class distribution.
- **Not addressing within-class imbalance:** SMOTE treats all minority class instances equally, which might not always be ideal. Within the minority class, some samples could be more challenging to classify than others.

- **Risk of overfitting:** By generating synthetic samples in the feature space, there is a risk of overfitting, especially if the synthetic samples are not diverse enough or too similar to the existing samples.

2.1.2 Other SMOTE Extensions

In this section, a variety of SMOTE-based techniques will be explored to address the challenges of class imbalance and abnormal instances in high-dimensional data, as summarized in the accompanying Table 2.1. The methods analyzed include BorderlineSMOTE (Han et al.; 2005), ADASYN (He et al.; 2008), SMOTE-LOF (Asniar et al.; 2022), SMOTE IPF (Sáez et al.; 2015), DBSMOTE (Bunkhumpornpat et al.; 2012), ProWSyn (Barua et al.; 2013), DSMOTE (Mahmoudi et al.; 2014), SMOTE TomekLinks (Batista et al.; 2004), SMOTE ENN (Batista et al.; 2004), Cluster SMOTE (Cieslak et al.; 2006), and Gaussian SMOTE (Lee and Kim; 2017). These algorithms differ in their initial sample selection strategies, ranging from nearest neighbours and clustering techniques to more advanced approaches like density peaks and LOF (Local Outlier Factor). Individual explanations of each method are provided in Section 5.2.

The effectiveness and nuances of these SMOTE variants have been extensively studied in the literature. For instance, in an empirical comparison of minority oversampling techniques, methods like BorderlineSMOTE and ADASYN have demonstrated particular effectiveness across different types of imbalanced datasets, underscoring the importance of selecting the appropriate method based on specific dataset characteristics (Kovács; 2019). Similarly, in a review by Fernandez et al. (2018), which marks the 15-year anniversary of SMOTE, the authors discuss the progress and challenges in the field, including the development of advanced SMOTE variants that incorporate filtering mechanisms (such as SMOTE IPF and SMOTE TomekLinks) and clustering-based approaches (such as DBSMOTE and Cluster SMOTE), which are crucial for handling more complex data distributions.

Table 2.1 summarizes key features of each selected re-sampling technique. 'Initial Selection' refers to the strategy of choosing samples for generating synthetic data. 'Sampling Method' indicates whether the method involves over-sampling, under-sampling, or both. 'Type of Interpolation' describes the technique used to generate synthetic samples. 'Adaptive Generation' shows if the method adapts to the data distribution during sample generation. 'Filtering' indicates if the method

filters out noisy or redundant data points where as 'Outliers Handling' details how the method manages outliers.

Table 2.1 summarizes how these methods integrate various types of interpolation techniques—such as linear, density-based, and Gaussian-based approaches—to generate synthetic data points. Some methods, like SMOTE IPF and SMOTE TomekLinks, incorporate filtering steps to remove noisy or redundant instances, while others, such as DBSMOTE and DSMOTE, specifically address outlier handling through clustering and density-based strategies. The adaptability of these methods to the underlying data distribution is also highlighted, with certain algorithms like ADASYN and DSMOTE offering adaptive generation of synthetic samples. Collectively, the techniques summarized in Table 2.1 offer a toolkit for addressing the complex challenges associated with imbalanced datasets, each with unique strengths tailored to specific scenarios encountered in high-dimensional data analysis.

Table 2.1: Categorization of SMOTE algorithm extensions

	Algorithm Name	Initial Selection	Sampling Method	Type of Interpolation	Adaptive Generation	Filtering	Outliers Handling
1	BorderlineSMOTE (Han et al.; 2005)	Nearest Neighbors	Over-sampling	Linear Interpolation on borderline samples			Indirect Handling via sampling on borderlines
2	ADASYN (He et al.; 2008)	Nearest Neighbors	Over-sampling	Density-based Interpolation	Yes		Indirect Handling via adaptive sampling
3	SMOTE-LOF (Asniar et al.; 2022)	Nearest Neighbors + LOF	Over-sampling	Linear Interpolation			Direct Handling via LOF
4	SMOTE IPF (Sáez et al.; 2015)	Nearest Neighbors	Over-sampling	Linear Interpolation			Indirect Handling via iterative filtering
5	DBSMOTE (Bunkhumpornpat et al.; 2012)	DBSCAN Clustering	Over-sampling	Graph based			Direct Handling via clustering
6	ProWSyn (Barua et al.; 2013)	Probability-weighted	Over-sampling	Weighted Interpolation	Yes		Indirect Handling via weighted sampling
7	DSMOTE (Mahmoudi et al.; 2014)	Density Peaks	Over-sampling	Density-based Interpolation	Yes		Direct Handling via density peaks
8	SMOTE TomekLinks (Batista et al.; 2004)	Nearest Neighbors	Over-sampling + Under-sampling	Linear Interpolation + Tomek Links	Yes	Yes	Indirect Handling via Tomek links
9	SMOTE ENN (Batista et al.; 2004)	Nearest Neighbors	Over-sampling + Under-sampling	Linear Interpolation + ENN		Yes	Indirect Handling via ENN
10	Cluster SMOTE (Cieslak et al.; 2006)	K-means clustering	Over-sampling	Cluster-based Interpolation			Indirect Handling via clustering
11	Gaussian SMOTE (Lee and Kim; 2017)	Nearest Neighbors	Over-sampling	Gaussian-based Interpolation			Indirect Handling via Gaussian noise

2.2 Local Outlier Factor (LOF)

Local Outlier Factor (LOF) used to detect anomalies in real-world applications in Chapters 5 and 6 (Breunig et al.; 2000; Duan et al.; 2007) is a well-known unsupervised data mining technique that identifies anomalous points within a dataset

without assuming any specific data distribution. Below is a brief summary of the LOF calculation process. Given a training dataset $\mathbf{X} \in \mathbb{R}^{n \times m}$ and test sample $x \in \mathbb{R}^m$, the LOF computation involves the following steps (Wang et al.; 2018):

1. Find the k Nearest Neighbors: For the sample x, identify its k nearest neibours using Euclidean distance, denoted as $\mathbf{N}(x) = [x_1, x_2, \ldots, x_k]$, where k is a predefined parameter representing the size of $\mathbf{N}(x)$.

2. Determine k distance: For each neighbor sample $x_f (1 \leq f \leq k)$, determine its k distance $k_distance(x_f)$, which is the Euclidean distance between x_f and its k^{th} nearest neighbor.

3. Compute Reachability Distance: Obtain the reachability distance of the sample x, defined as:

$$reach_dist(x, x_f) = max\{k_distance(x_f), d(x, x_f)\}$$

   where $d(x, x_f)$ is the Euclidean distance between x and its f^{th} neighbor.

4. Compute Local Reachability Density (lrd): Calculate the local reachability density for the sample x as follows:

$$lrd(x) = \frac{k}{\sum_{f=1}^{k} reach_dist(x, x_f)}.$$

5. Compute Local Outlier Factor (LOF): Compute the LOF of the sample x by:

$$LOF(x) = \frac{1}{k} \sum_{f=1}^{k} \frac{lrd(x_f)}{lrd(x)}.$$

From the LOF calculation procedure, it is evident that the LOF value reflects the extent to which a sample is isolated relative to its neighboring points. If a sample x is not an outlier, its LOF value will be approximately 1. Conversely, if x is an outlier, the LOF value will be greater than 1, as the local reachability density (lrd) of x will be lower compared to its neighbors. Thus, LOF effectively identifies outliers without relying on any specific data distribution assumptions.

2.3 Dimensionality Reduction

2.3.1 The Curse of Dimensionality

High dimensionality presents a dual-edged nature in data analysis, often referred to as both the blessing and the curse of dimensionality. The curse of dimensionality, a concept introduced by Bellman (1957), broadly encompasses the challenges of dealing with high-dimensional data (Altman and Krzywinski; 2018). As dimensionality increases, data points become sparse (Hastie et al.; 2009; Zimek et al.; 2012), creating vast empty spaces between them and complicating the identification of meaningful patterns and reliable statistical estimates. This sparsity also exacerbates the computational inefficiency of methods like grid search, as the volume of the search space expands exponentially with each additional dimension, making exhaustive search methods impractical. Additionally, high-dimensional datasets frequently suffer from multicollinearity (Aguilera et al.; 2006; Krzywinski and Altman; 2015), where features are highly correlated, leading to instability in model coefficients and making it difficult to assess the true impact of each variable. The issue of multiple testing also arises (Krzywinski and Altman; 2014), where the increased number of features leads to a higher likelihood of false positives, complicating the interpretation of results. Moreover, high-dimensional models are prone to overfitting (Lever et al.; 2016), as they tend to capture noise in the data rather than the underlying signal, resulting in poor generalization to new data.

Conversely, the "blessing of dimensionality" refers to the advantageous aspects of high-dimensional spaces. Despite the challenges, high dimensions can offer increased flexibility and more distinct separation of data points, potentially leading to more powerful and precise models. Both the blessing and the curse stem from the same mathematical principles, illustrating that while high dimensionality can enhance data representation and model performance, it also introduces significant computational and analytical complexities (Donoho et al.; 2000). The intuitive notion underpinning these concepts is that "there is too much room in higher dimensional spaces," which can be both a benefit and a burden in data analysis and optimization tasks.

In Chapter 6, I further explore the curse of dimensionality and how dimensionality reduction techniques like Principal Component Analysis (PCA) and Autoencoders address this challenge. PCA reduces dimensionality through linear projection, maximizing variance, while Autoencoders, a type of neural network, offer a nonlinear approach by encoding data into a latent space. These methods are crucial for simplifying complex, high-dimensional datasets while retaining essential information.

2.3.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) (Hotelling; 1933) is a dimensionality reduction technique that identifies the directions (principal components) along which the variance of a dataset is maximized. This method transforms a high-dimensional dataset into a lower-dimensional space, simplifying the data while preserving its most significant features. PCA achieves this by projecting the original data onto a new set of orthogonal axes, capturing the largest amount of variance with the fewest number of components. This projection allows for a more manageable representation of complex datasets, making it easier to visualize and interpret. Widely used in fields like pattern recognition and computer vision, including applications such as face recognition, PCA is valued for its ability to reduce dimensionality and enhance the clarity of data representation. As a fundamental and widely adopted technique, PCA plays a crucial role in simplifying high-dimensional data and facilitating more effective analysis.

2.3.3 Autoencoders

Autoencoders (Goodfellow et al.; 2016) have emerged as versatile tools for feature reduction in neural networks, facilitating the extraction of essential information from high-dimensional datasets. Combining an encoder and a decoder, autoencoders compress input data into a lower-dimensional representation, capturing the most salient features while discarding redundant information. In the context of neural network architectures, autoencoders serve as a means to enhance the efficiency and performance of classification models. By extracting encoded representations from the learned encoder, autoencoders enable significant reductions in data dimensionality, thereby streamlining the processing and analysis of complex datasets. Leveraging these encoded representations as input features empowers

classification models to achieve improved accuracy and generalization, particularly in scenarios characterized by high-dimensional data or limited labelled samples.

Using autoencoders as a feature reduction method offers a promising avenue for advancing machine learning research and applications. This approach enhances efficiency of classification models and broadens the applicability of machine learning techniques across diverse domains, where effective feature reduction is paramount for accurate and robust data analysis.

2.4 Important Classifiers

2.4.1 Logistic Regression

Logistic regression (Weisberg; 2005) is widely used to model the relationship between dichotomous dependent variables and one or more continuous or categorical explanatory variables. Consider a training data set with n samples and m explanatory variables, $x_1, \ldots, x_m$. The logistic regression can be made by,

$$h_{(b,w)}(x) = \frac{1}{1 + e^{-(b+w^T x)}}$$

where b is the model intercept, w represents coefficients $w_1, \ldots, w_m$ and $w^T x = w_1 x_1 + \ldots + w_m x_m$.

Suppose the response variable takes $Y = 0, 1$ values. Thus,

$$P(y = 1|x; (b, w)) = h_{(b,w)}(x)$$
$$P(y = 0|x; (b, w)) = 1 - h_{(b,w)}(x)$$

And combining them, we obtain,

$$P(y|x; (b, w)) = h_{(b,w)}(x)^y (1 - h_{(b,w)}(x))^{1-y}$$

Hence, $b, w_1, \ldots, w_m$ can be estimated through the Maximum Likelihood Estimation (MLE) method. The likelihood function is:

$$L(b, w) = \prod_{i=1}^{n} p(y^{(i)}|x^{(i}; (b, w))$$

The log-likelihood function is:

$$\begin{aligned} l(b, w) &= ln(\prod_{i=1}^{n} p(y^{(i)}|x^{(i)}; (b, w))) \\ &= ln(\prod_{i=1}^{n} h_{(b,w)}(x^{(i)})^{y^{(i)}}(1 - h_{(b,w)}(x^{(i)}))^{1-y^{(i)}}) \\ &= \sum_{i=1}^{n} y^{(i)} ln(h_{(b,w)}(x^{(i)})) + (1 - y^{(i)} ln(1 - h_{(b,w)}(x^{(i)}))) \end{aligned}$$

The goal is to find the set of parameter values that maximize this log-likelihood of observing the given set of outcomes. This estimation process involves iteratively adjusting the parameter values until convergence is achieved.

2.4.2 Logistic Regression with Elastic Net Regularization

Elastic Net classification (Zou and Hastie; 2005) is a statistical method used for classification problems, which combines the advantages of Lasso and Ridge regression methods. This method is based on a linear combination of L1 and L2 regularization penalties, which are applied to the coefficients of the linear classification model. The L1 penalty encourages sparsity in the coefficients, leading to feature selection and improved interpretability, while the L2 penalty helps to reduce overfitting by shrinking the coefficients towards zero. The Elastic Net method can be solved using various optimization algorithms, such as coordinate descent, proximal gradient descent, and stochastic gradient descent. The algorithm's choice depends on the dataset's size and the solution's sparsity. Here, for this analysis, we used the "proximal Newton" algorithm for the optimization criterion for fast convergence. This makes repeated use of a quadratic approximation to the log-likelihood and then weighted coordinate descent on the resulting penalized weighted least-squares problem. These constitute an outer and inner loop known as iteratively reweighted penalized least squares. Elastic net regularization seeks to find coefficients that can minimize

$$\min_{(b,w)\in\mathbb{R}^{m+1}} \left(-\frac{l(b,w)}{n} + \lambda P_\alpha(w) \right)$$

where, $P_\alpha(w) = (1-\alpha)\frac{1}{2}\|w\|_2^2 + \alpha\|w\|_1$. Also,

$$\|w\|_1 = \sum_{j=1}^{p} |w_j|, \qquad \text{and} \qquad \|w\|_2 = \left(\sum_{j=1}^{p} w_j^2 \right)^{\frac{1}{2}}$$

Here $\lambda \geq 0$ and $\alpha \in [0,1]$ are hyper-parameters that control the strength of the L1 and L2 penalties, respectively. The elastic net penalty is controlled by α, and bridges the gap between lasso regression ($\alpha = 1$) and ridge regression ($\alpha = 0$). The tuning parameter λ controls the overall strength of the penalty.

2.4.3 k-Nearest Neighbors (k-NN) Classifier

The k-Nearest Neighbors (k-NN) classifier is a simple, yet effective, non-parametric method used for classification and regression tasks. Its strength lies in its simplicity and the intuitive way it classifies new data points based on the majority class among its nearest neighbors.

The k-NN algorithm operates by finding the k-nearest neighbours of a query point and classifying the point based on the majority vote of these neighbours. The k-NN algorithm is:

Algorithm 1 k-Nearest Neighbors (k-NN) Classification

Input: Training data $\{(x_i, y_i)\}_{i=1}^N$ with feature vectors x_i and class labels y_i; Test data point x; Number of neighbors k; Distance metric $d(x, x_i)$.
Output: Predicted class label for x.

Step 1: Find Nearest Neighbors
1. Compute distances between x and all training points x_i using $d(x, x_i)$.
2. Identify the k training points with the smallest distances to x.

Step 2: Assign Class Label
3. Count the frequency of each class label among the k nearest neighbors.
4. Assign the most frequent class label $\hat{y}$ to the test point x.

The choice of distance metric significantly impacts the performance of the k-NN classifier. The most commonly used distance metrics include:

- **Euclidean Distance:** The Euclidean distance between two points x_i and x_j in n-dimensional space is given by:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{n} (x_{il} - x_{jl})^2} \tag{2.1}$$

- **Manhattan Distance:** The Manhattan distance, also known as the L_1 distance, is given by:

$$d(x_i, x_j) = \sum_{l=1}^{n} |x_{il} - x_{jl}| \tag{2.2}$$

- **Minkowski Distance:** A generalization of the Euclidean and Manhattan distances, the Minkowski distance of order p is defined as:

$$d(x_i, x_j) = \left(\sum_{l=1}^{n} |x_{il} - x_{jl}|^p \right)^{\frac{1}{p}} \tag{2.3}$$

For $p = 2$, the Minkowski distance becomes the Euclidean distance, and for $p = 1$, it becomes the Manhattan distance.

The choice of k is critical in the k-NN algorithm. A small k can lead to a noisy classification, while a large k may smooth out the classification too much, leading to underfitting. A common practice is to use cross-validation to determine the optimal value of k.

2.4.4 Random Forest Classifier

The Random Forest classifier (Breiman; 2001) is a robust ensemble learning method that builds multiple decision trees and merges their results to enhance classification accuracy and control overfitting. Each tree is trained on a bootstrap sample of the training data, and the final classification decision is made by aggregating the predictions of all individual trees, typically through majority voting for classification problems. This approach leverages multiple weak learners to form a strong overall model.

Each decision tree in the Random Forest is constructed using the following steps:

1. **Bootstrap Sampling:** A random sample with replacement (bootstrap sample) is drawn from the original training data. This sample is used to train an individual tree.

2. **Random Feature Selection:** At each node of the tree, a random subset of features is selected. The best feature from this subset is used to split the node.

3. **Tree Growth:** The tree is grown to its maximum depth without pruning. Each node splits based on the best feature (from the random subset) that minimizes a chosen impurity measure of the class labels, such as Gini impurity or entropy.

The quality of a split at each node is evaluated using an impurity measure. Two common impurity measures are:

- **Gini Impurity:** The Gini impurity of a node is given by:

$$G = 1 - \sum_{i=1}^{C} p_i^2$$

  where p_i is the proportion of instances belonging to class i at the node, and C is the total number of classes.

- **Entropy:** The entropy of a node is given by:

$$H = -\sum_{i=1}^{C} p_i log_2(p_i)$$

  Entropy measures the disorder or uncertainty at a node, with higher values indicating greater impurity.

$$\hat{y} = \underset{C}{argmax} \sum_{t=1}^{T} \mathbb{I}(y_t = C) \tag{2.4}$$

where:

- $\hat{y}$ is the predicted class label for the new data point.
- T is the total number of trees in the forest.
- $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the argument is true and 0 otherwise.
- y_t is the predicted class label from the t^{th} tree.

One of the advantages of the Random Forest classifier is its ability to provide an estimate of feature importance. This is typically done by measuring the total decrease in node impurity (e.g., Gini impurity) brought by each feature across all trees in the forest. Features that result in larger impurity decreases are considered more important.

However, both Random Forest and k-Nearest Neighbors (k-NN) are inherently sensitive to class imbalance due to their reliance on majority voting, which can bias the model towards the majority class.

2.5 Performance Measures

To assess the performance of classifiers developed using imbalanced data, several measures have been proposed in the literature (He and Garcia; 2009). The commonly used metric, total accuracy (Eq. 2.5), does not account for the differences in correct classifications across various classes, leading to potentially misleading interpretations in imbalanced scenarios in binary classification.

$$\text{Total Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

where,

- TP (True Positives) are the instances correctly classified as positive.
- TN (True Negatives) are the instances correctly classified as negative.
- FP (False Positives) are the instances incorrectly classified as positive.
- FN (False Negatives) are the instances incorrectly classified as negative.

Therefore, this research uses a range of alternative accuracy measures, including the F1 Score, Recall, Precision, MCC, and PR-AUC, to provide a more comprehensive evaluation of the classifiers' effectiveness.

Confusion Matrix

A confusion matrix provides a comprehensive view of the classifier's performance by showing the counts of true positives, true negatives, false positives, and false negatives. It allows for the calculation of all the aforementioned metrics and provides insights into specific types of errors the classifier makes.

Table 2.2: Confusion Matrix

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

1. **Precision and Recall**

   Precision and recall are crucial metrics for evaluating a classifier's performance, especially in imbalanced datasets. Precision, also known as positive predictive value, measures the accuracy of the positive predictions.

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

   Recall, also known as sensitivity, measures the ability of the classifier to identify all positive instances.

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

   Precision and recall often exhibit a trade-off; improving one typically reduces the other. Therefore, they are often used in conjunction to provide a more comprehensive performance evaluation.

2. **F1 Score**

   The F1 score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is particularly useful when dealing

with imbalanced datasets, where focusing solely on accuracy can be misleading.

$$\text{F1 Score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The F1 score ranges from 0 to 1, with 1 indicating perfect precision and recall.

3. **Precision-Recall Curve and Area Under the Curve (PR AUC)** The Precision-Recall (PR) curve is particularly useful for imbalanced datasets, where the focus is on the minority class. It plots precision (positive predictive value) against recall (sensitivity) at various threshold settings [1]. The area under the PR curve (PR AUC) provides a single scalar value to summarize the performance, similar to ROC AUC (Hanley and McNeil; 1982). PR AUC is often more informative than ROC AUC in cases of severe class imbalance, as it highlights the trade-offs between precision and recall without being influenced by the majority class. $PRAUC = 1$ is ideal and .5 is no better than random guess.

4. **Matthews Correlation Coefficient (MCC)** The MCC is a correlation coefficient between the observed and predicted binary classifications, returning a value between -1 and $+1$. An MCC of $+1$ indicates perfect prediction, 0 indicates no better than random prediction, and -1 indicates total disagreement between prediction and observation.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

The MCC is particularly useful in scenarios with imbalanced classes, as it considers all elements of the confusion matrix.

Bibliography

Aguilera, A. M., Escabias, M. and Valderrama, M. J. (2006). Using principal components for estimating logistic regression with high-dimensional multicollinear

[1] Thresholds in Logistic Regression are the probability cutoffs used to classify instances as positive or negative, influencing the trade-off between precision and recall on the PR curve.

data, *Computational Statistics & Data Analysis* **50**(8): 1905–1924.
URL: *https://www.sciencedirect.com/science/article/pii/S0167947305000630*

Altman, N. and Krzywinski, M. (2018). The curse(s) of dimensionality, *Nature methods* **15**(6): 399–400.

Asniar, Maulidevi, N. U. and Surendro, K. (2022). SMOTE-LOF for noise identification in imbalanced data classification, *Journal of King Saud University. Computer and information sciences* **34**(6): 3413–3423.

Barandela, R., Sanchez, J., Garcia, V. and Rangel, E. (2003). Strategies for learning in class imbalance problems, *Pattern recognition* **36**(3): 849–851.

Barua, S., Islam, Md, M. and Murase, K. (2013). Prowsyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning, *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, pp. 317–328.

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

Bellman, R. (1957). *Dynamic programming.*, Princeton University Press, Princeton.

Breiman, L. (1996). Bagging predictors, *Machine learning* **24**(2): 123–140.

Breiman, L. (2001). Random forests, *Machine learning* **45**(1): 5–32.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: identifying density-based local outliers, *International Conference on Management of Data: Proceedings of the 2000 ACM SIGMOD international conference on Management of data : Dallas, Texas, United States; 15-18 May 2000*, ACM, New York, NY, USA, pp. 93–104.

Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C. (2012). DBSMOTE: Density-based synthetic minority over-sampling technique, *Applied intelligence (Dordrecht, Netherlands)* **36**(3): 664–684.

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

Chawla, N. V., Japkowicz, N. and Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets, *SIGKDD explorations* **6**(1): 1–6.

Chawla, N. V., Lazarevic, A., Hall, L. O. and Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting, *Knowledge Discovery in Databases: PKDD 2003*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 107–119.

Cieslak, D., Chawla, N. and Striegel, A. (2006). Combating imbalance in network intrusion datasets, *2006 IEEE International Conference on Granular Computing*, pp. 732–737.

Dai, D. and Hua, S. (2016). Random under-sampling ensemble methods for highly imbalanced rare disease classification, *Proceedings of the International Conference on Data Mining (DMIN)* .

Donoho, D. L. et al. (2000). High-dimensional data analysis: The curses and blessings of dimensionality, *AMS math challenges lecture* **1**(2000): 32.

Duan, L., Xu, L., Guo, F., Lee, J. and Yan, B. (2007). A local-density based spatial clustering algorithm with noise, *Information systems (Oxford)* **32**(7): 978–986.

Elkan, C. P. (2001). The foundations of cost-sensitive learning, *International Joint Conference on Artificial Intelligence*.
URL: *https://api.semanticscholar.org/CorpusID:16149383*

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

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

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets, *International Journal of Emerging Technology and Advanced Engineering* **2**: 42–47.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning*, Adaptive computation and machine learning, The MIT Press, Cambridge, Massachusetts.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning, *Advances in Intelligent Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 878–887.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve, *Radiology* **143**(1): 29–36.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, second edition edn, Springer, New York.

He, H., Bai, Y., Garcia, E. and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning, *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Vol. 10, IEEE, pp. 1322–1328.

He, H. and Garcia, E. (2009). Learning from imbalanced data, *IEEE transactions on knowledge and data engineering* **21**(9): 1263–1284.

Hotelling, H. (1933). *Analysis of a complex of statistical variables into principal components*, Warwick York, Baltimore.

Huang, K., Yang, H., King, I. and Lyu, M. (2006). Imbalanced learning with a biased minimax probability machine, *IEEE transactions on systems, man and cybernetics. Part B, Cybernetics* **36**(4): 913–923.

Ileberi, E., Sun, Y. and Wang, Z. (2021). Performance evaluation of machine learning methods for credit card fraud detection using smote and adaboost, *IEEE access* **9**: 165286–165294.

Jiang, X. and Ge, Z. (2021). Data augmentation classifier for imbalanced fault classification, *IEEE Transactions on Automation Science and Engineering* **18**(3): 1206–1217.

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

Krzywinski, M. and Altman, N. (2014). Comparing samples—part ii, *Nature methods* **11**(4): 355–356.

Krzywinski, M. and Altman, N. (2015). Multiple linear regression, *Nature methods* **12**(12): 1103–1104.

Kumaran, S. Y., Jeya, J. J., Mahesh, T. R., Khan, S. B., Alzahrani, S. and Alojail, M. (2024). Explainable lung cancer classification with ensemble transfer learning of vgg16, resnet50 and inceptionv3 using grad-cam, *BMC medical imaging* **24**(1): 176–19.

Lee, Hansoo; Kim, J. and Kim, S. (2017). Gaussian-based smote algorithm for solving skewed class distributions, *Int. J. Fuzzy Log. Intell. Syst.* **17**: 229–234.

Lee, W., Jun, C.-H. and Lee, J.-S. (2017). Instance categorization by support vector machines to adjust weights in adaboost for imbalanced data classification, *Information Sciences* **381**: 92–103.
URL: *https://www.sciencedirect.com/science/article/pii/S0020025516318047*

Lever, J., Krzywinski, M. and Altman, N. (2016). Points of significance: Model selection and overfitting, *Nature methods* **13**(9): 703–.

Li, J., Fong, S., Hu, S., Chu, V. W., Wong, R. K., Mohammed, S. and Dey, N. (2017). Rare event prediction using similarity majority under-sampling technique, *Soft Computing in Data Science*, Communications in Computer and Information Science, Springer Singapore, Singapore, pp. 23–39.

Mahmoudi, S., Moradi, P., Akhlaghian, F. and Moradi, R. (2014). Diversity and separable metrics in over-sampling technique for imbalanced data classification, *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 152–158.

Saini, M. and Susan, S. (2020). Deep transfer with minority data augmentation for imbalanced breast cancer dataset, *Applied Soft Computing* **97**: 106759.
URL: *https://www.sciencedirect.com/science/article/pii/S1568494620306979*

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J. and Napolitano, A. (2010). Rusboost: A hybrid approach to alleviating class imbalance, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **40**(1): 185–197.

Sisodia, D. S., Reddy, N. K. and Bhandari, S. (2017). Performance evaluation of class balancing techniques for credit card fraud detection, *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, IEEE, pp. 2747–2752.

Sun, Y., Kamel, M. S., Wong, A. K. and Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data, *Pattern recognition* **40**(12): 3358–3378.

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 re-sampling method with filtering, *Information sciences* **291**: 184–203.

Wang, L., Deng, X. and Cao, Y. (2018). Multimode complex process monitoring using double-level local information based local outlier factor method, *Journal of chemometrics* **32**(10).

Weisberg, S. (2005). *Applied Linear Regression*, John Wiley and Sons, Incorporated, Hoboken.

Wong, S. C., Gatt, A., Stamatescu, V. and McDonnell, M. D. (2016). Understanding data augmentation for classification: When to warp?, *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–6.

Zimek, A., Schubert, E. and Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data, *Statistical analysis and data mining* **5**(5): 363–387.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society. Series B, Statistical methodology* **67**(2): 301–320.

Chapter 3

Discovering Long COVID Symptom Patterns: Association Rule Mining and Sentiment Analysis in Social Media Tweets

The content of this chapter is from the publication cited as follows:

Matharaarachchi, S., Domaratzki, M., Katz, A. and Muthukumarana, S. (2022). Discovering long COVID symptom patterns: Association rule mining and sentiment analysis in social media tweets, *JMIR formative research* 6(9): e37984–e37984.

Discovering Long COVID Symptom Patterns: Association Rule Mining and Sentiment Analysis in Social Media Tweets

Surani Matharaarachchi[1], BSc, MSc; Mike Domaratzki[2], BMATH, MMATH, PhD; Alan Katz[3], MBChB, MSc, CCFP; Saman Muthukumarana[1], BSc, MSc, PhD

[1]Department of Statistics, University of Manitoba, Winnipeg, MB, Canada
[2]Department of Computer Science, Western University, London, ON, Canada
[3]Department of Community Health Sciences and Family Medicine, University of Manitoba, Winnipeg, MB, Canada

Corresponding Author:
Surani Matharaarachchi, BSc, MSc
Department of Statistics
University of Manitoba
66 Chancellors Cir
Winnipeg, MB, R3T 2N2
Canada
Phone: 1 431 338 5077
Email: matharas@myumanitoba.ca

Abstract

Background: The COVID-19 pandemic is a substantial public health crisis that negatively affects human health and well-being. As a result of being infected with the coronavirus, patients can experience long-term health effects called long COVID syndrome. Multiple symptoms characterize this syndrome, and it is crucial to identify these

symptoms as they may negatively impact patients' day-to-day lives. Breathlessness, fatigue, and brain fog are the 3 most common continuing and debilitating symptoms that patients with long COVID have reported, often months after the onset of COVID-19.

Objective: This study aimed to understand the patterns and behavior of long COVID symptoms reported by patients on the Twitter social media platform, which is vital to improving our understanding of long COVID.

Methods: Long COVID–related Twitter data were collected from May 1, 2020, to December 31, 2021. We used association rule mining techniques to identify frequent symptoms and establish relationships between symptoms among patients with long COVID in Twitter social media discussions. The highest confidence level–based detection was used to determine the most significant rules with 10% minimum confidence and 0.01% minimum support with a positive lift.

Results: Among the 30,327 tweets included in our study, the most frequent symptoms were brain fog (n=7812, 25.8%), fatigue (n=5284, 17.4%), breathing/lung issues (n=4750, 15.7%), heart issues (n=2900, 9.6%), flu symptoms (n=2824, 9.3%), depression (n=2256, 7.4%) and general pains (n=1786, 5.9%). Loss of smell and taste, cold, cough, chest pain, fever, headache, and arm pain emerged in 1.6% (n=474) to 5.3% (n=1616) of patients with long COVID. Furthermore, the highest confidence level–based detection successfully demonstrates the potential of association analysis and the apriori algorithm to establish patterns to explore 57 meaningful relationship rules among long COVID symptoms. The strongest relationship revealed that patients with lung/breathing problems and loss of taste are likely to have a loss of smell with 77% confidence.

Conclusions: There are very active social media discussions that could support the growing understanding of COVID-19 and its long-term impact. These discussions enable a potential field of research to analyze the behaviour of long COVID syndrome. Exploratory data analysis using natural language processing methods revealed the symptoms and medical conditions related to long COVID discussions on the Twitter social media platform. We determined interesting and meaningful relationships between symptoms using apriori algorithm- based association rules.

Keywords: *COVID-19; long COVID symptoms; social media analysis; association rule mining; bigram analysis; natural language processing; Twitter; content analysis; data mining; infodemiology; health information*

3.1 Introduction

The Coronavirus disease (COVID-19), a transmissible disease caused by the SARS-CoV-2 virus, has become a significant public health crisis that negatively affects people's health and well-being. Most people with COVID-19 recover entirely within weeks. But some people still experience symptoms after their initial recovery, even those who had mild symptoms with their initial infection. Others develop new symptoms related to their COVID illness. These people sometimes describe themselves as "long haulers" (Callard and Perego; 2021). This syndrome has been described as post-COVID-19 or "long COVID-19" (Yong; 2022). It is crucial to identify these symptoms as they may negatively impact those inflicted's day-to-day lives for a significantly long period. Breathlessness, fatigue, and brain fog are the three most common symptoms that long COVID patients have consistently reported, often months after the onset of the COVID-19 disease (Editorial; 2020; Soriano et al.; 2022).

Social media has become a significant part of our lives. People use it to connect with others and share their thoughts, emotions, and experiences about any current topic, often without revealing their identity (Siddiqui and Singh; 2016). Hence, social media platforms such as Twitter®, Facebook®, and Instagram® have had a massive impact on society and gained considerable research attention (Fitzgerald and Radmanesh; 2015). There are extensive chains of discussions on these platforms about the long COVID syndrome (LCS). Thus, analyzing social media conversations of long COVID-related patients from various sources provides an opportunity to understand the relationship between symptoms and their consequences.

According to the World Health Organization clinical case definition (Soriano et al.; 2022)," long COVID-19 condition occurs in individuals with a history of probable or confirmed SARS CoV2 infection, usually three months from the onset of COVID-19, with symptoms that last for at least two months." Health-related organizations, such as Mayo clinic (Joshee et al.; 2022), the National Health Service (NHS) (NHS; 2022), Centers for Disease Control and Prevention (CDC) (for Disease Control and Prevention; 2022), and World Health Organization (Soriano et al.; 2022), have identified different lists of symptoms related to long COVID and a summary is presented in Table 3.2. The last column of Table 3.2 presents the results of recent research that has been conducted using self-reported long COVID symptoms on Twitter (Reddy et al.; 2023). These symptoms were identified by manually reading 165 tweets from 20 July 2020 to 29 July 2020.

There are, however, very active ongoing social media discussions that could support the growing understanding of the illness and its long-term impact. This provides the opportunity to access publicly available data from multiple individuals on Twitter to

analyze long COVID symptoms. However, manually discovering the knowledge in a large volume of unstructured texts is increasingly problematic. Hence automated Natural Language Processing (NLP) methods have been introduced to do this job effectively and accurately (Allen; 1995; Chowdhary; 2020). The significance of using NLP methods in health sciences has been increasingly recognized over recent years (Chen et al.; 2018). Even though previous research has identified a list of symptoms, extracting and grouping the symptoms discussed in tweets into categories, makes them easier to analyze and to find the relationships between the most common symptoms. Manually handling this task is challenging; thus, NLP tools represent an opportunity to extract such hidden information from unstructured Twitter text data.

Association rules are considered a useful tool as they offer the possibility to conduct intelligent diagnoses, extract invaluable information, and build important knowledge quickly and automatically while identifying relationships within and between variables (Stilou et al.; 2001). Thus, we used Association Rule Mining (ARM) to discover relationships among long COVID symptoms based on symptoms revealed from text data (Bird and Klein; 2009). The mining process of association rules requires setting a minimum confidence and support threshold to describe association rules that are meaningful. Therefore, we identified association rules based on a minimum confidence threshold of 10% and a minimum support threshold of 0.1%.

The study sought to achieve the two goals. The first one is to identify the symptoms and medical conditions related to long COVID that were discussed on the Twitter social media platform. The second goal is to determine the patterns of symptoms and their associations. By accomplishing these objectives, this work will ultimately help physicians identify the behavior of the long COVID patients. This paper provides new ideas for symptom mining and reveals the internal relationship between symptoms and their application value. Thus, the work has theoretical and practical significance.

3.2 Methods and Experimental Design

3.2.1 Data Collection

We collected worldwide long COVID-related English tweets between 1 May 2020 and 31 December 2021 to create our data set of about 1M tweets. We used the Snscrape module in Python 3.8 (Argamon and Lee; 2009) to scrape the tweet text online from tweets that match the keyword "LongCovid." Streaming English tweets from multilingual tweets is computationally intensive. Because most non-native English speakers have used their native languages rather than English to express their feelings on social media, it

requires translating each tweet to English if we wish to analyze them. Therefore, we limited the data set to English tweets. The most useful attributes of our data set are id [Number], created at [DateTime], original text [Text], and language [Text], which we used to filter English tweets only. We removed duplicates of tweet text from our dataset. This processing necessarily removed any retweets in our dataset, which is consistent with our goals of collecting a dataset that represents patient experiences. The average tweet length of the initial data set was 32.56 words.

We reduced the data set to 127,848 tweets by limiting the population to those who suffered from COVID-19. To do this, we refined the tweets to ensure that all the tweets reflect personal experiences with long COVID. We first considered tweets containing the pronoun 'I' and the word 'covid' as we want to extract tweets from people with COVID/long COVID. Then, we removed tweets containing words that explain users' opinions as many people discuss long COVID without necessarily having COVID. The set of such words/phrases we considered is listed in Table 3.1 with the percentage of tweets including the specific word. The phrase 'I feel' may include tweets that express the experience of symptoms; however, to remove the possibility of opinion, we have taken these tweets out. We also observed a number of tweets (2.28%) talking about similar conditions, especially chronic fatigue syndrome (CFS). To eliminate the context related to CFS, we removed tweets containing the keyword 'cfs.' Our data set was further reduced to identify only the patients discussing their experiences with the symptoms. First, we created a list of long COVID symptoms discussed in different literature sources, as shown in Table 3.2. Then, we found the most common symptoms by comparing them with the pre-processed word corpus shown in Table 3.3.

Table 3.1: The list of words explains users' opinions rather than the experience.

Word/ phrase	Percentage of Tweets
'opinion'	0.43
'i believe'	0.8
'i think'	4.61
'i feel'	1.35
'may be'/ 'maybe'/ 'might'	5.1
'perhaps'	0.5

Table 3.2: List of long COVID symptoms identified by different sources in the literature.

	Symptom	Mayoclinic	NHS	CDC	WHO	Singh and Reddy (10)
1	Extreme tiredness (Fatigue)	✓	✓	✓	✓	✓
2	Shortness of breath or difficulty breathing	✓	✓	✓	✓	✓
3	Cough	✓	✓	✓	✓	✓
4	Joint pain	✓	✓	✓	✓	
5	Chest pain or tightness	✓	✓	✓	✓	✓
6	Problems with memory and concentration ("brain fog")	✓	✓	✓	✓	✓
7	Difficulty sleeping (insomnia)		✓	✓	✓	✓
8	Muscle pain	✓		✓	✓	✓
9	Headache	✓	✓	✓	✓	✓
10	Fast or pounding heartbeat (Heart palpitations)/ Tachycardia	✓	✓	✓	✓	✓
11	Loss of smell	✓	✓	✓	✓	✓
12	Loss of taste	✓	✓	✓	✓	✓
13	Depression or anxiety	✓				
14	Fever	✓	✓		✓	
15	Dizziness (light-headedness)	✓	✓	✓	✓	✓
16	Worsened symptoms after physical or mental activities	✓		✓		
17	Pins-and-needles feeling		✓	✓	✓	
18	Tinnitus, Earaches		✓		✓	✓
19	Diarrhea		✓	✓	✓	
20	Stomach aches		✓	✓		
21	Loss of appetite		✓			✓
22	Sore throat		✓			
23	Rash			✓		
24	Mood changes			✓		
25	Changes in menstrual period cycles			✓	✓	
26	Abdominal pain				✓	✓
27	Neuralgias				✓	
28	Allergies				✓	
29	Body pain					✓
30	Nausea					✓
31	Weakness					✓
32	Numbness					✓

Table 3.3: Pre-processed word corpus of stemmed symptoms.

Group	Symptoms
brain fog	'brain fog', 'brain', 'fog', 'memori', 'mental', 'rememb', 'concentr', 'mind', 'remind', 'focus'
fatigue	'fatigu', 'tire', 'exhaust'
lung	'lung', 'breathless', 'breath'
cant walk	'cant walk', 'struggl walk', 'unabl walk', 'couldnt walk', 'bare walk', 'unaid walk', 'stair walk'
depress	'depress', 'mood', 'stress', 'anxieti'
lose weight	'lose weight', 'loss weight'
insomnia	'cant sleep', 'insomnia'
diarrhea	'diarrhea', 'diarrhoea'
dizziness	'dizz', 'lighthead'
heart	'heart', 'heart palpit', 'tachycardia', 'dysautonomia', 'arrhythmia'
	'headach', 'neck', 'arm', 'muscl pain', 'cough', 'chest pain', 'flu', 'joint pain', 'pain', 'rash', 'fever', 'loss smell', 'loss tast', 'cold', 'earach', 'vomit', 'chill', 'nausea', 'faint', 'gain weight', 'trauma', 'bodi', 'bleed', 'appetit', 'sore throat', 'pin needl', 'numb', 'tinnitu', 'buzz', 'hairfall', 'nose', 'stomach', 'menstrual', 'abdomin'

3.2.2 Data Pre-Processing

Data pre-processing is mainly used to clean the raw data by following specific steps to achieve better results for further evaluations. We pre-processed our initial data to ensure quality by developing a user-defined pre-processing function based on NLTK (Natural Language Toolkit, a Python library for NLP) (Bird and Klein; 2009).

The pre-processing plan was as follows: First, we removed the hashtag symbol and its content (e.g., #COVID19, @users, and URLs) from the texts because the hashtag symbols or the URLs did not contribute to the text analysis. We also removed all non-English characters (non-ASCII characters) because the study focused on analyzing tweets in English. We then removed repeated words and stop-words identified by NLTK. Special characters, punctuation, and numbers from the data set were also removed as they did not help with detecting the profanity comments. We conducted a sentiment analysis to gauge how patients feel about long COVID-related topics, and the results are discussed in the next section. The primary focus of the study was symptom mining; therefore,

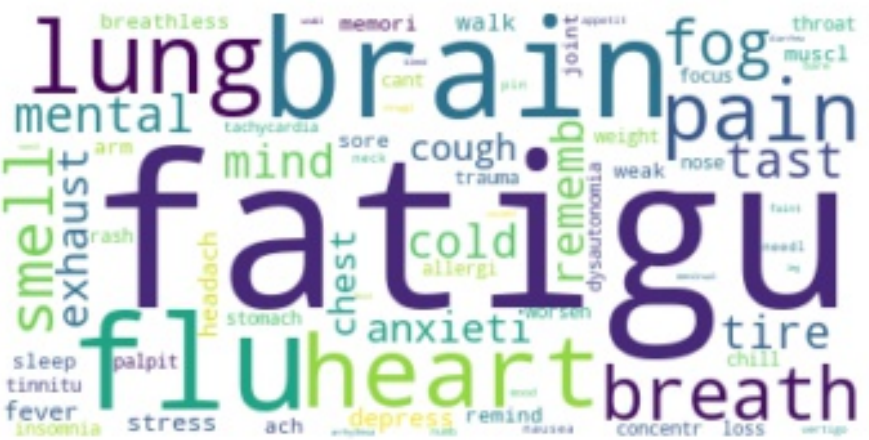


Figure 3.1: Word cloud of the list of 73 words after stemming.

we created a data set for patients with symptom information using a pre-defined set of keywords. Figure 3.1 shows the word cloud of the set of keywords accounted for initial tweets extraction.

Stemming reduces inflected words to their word stem, base, or root form, whereas tokenization is used to split each sentence into smaller parts of a word. Figure 3.2 shows how the data collection, cleaning, and pre-processing steps are conducted and how the final data set with 30,327 tweets is obtained for analysis in the study. After cleaning and pre-processing steps, the average tweet length was reduced from 32.56 to 19.07 words. Initially obtained data (original data) and the data considered for the study (filtered data) have been plotted in the same time series plot (Figure 3.3). Filtered data represent the original data appropriately.

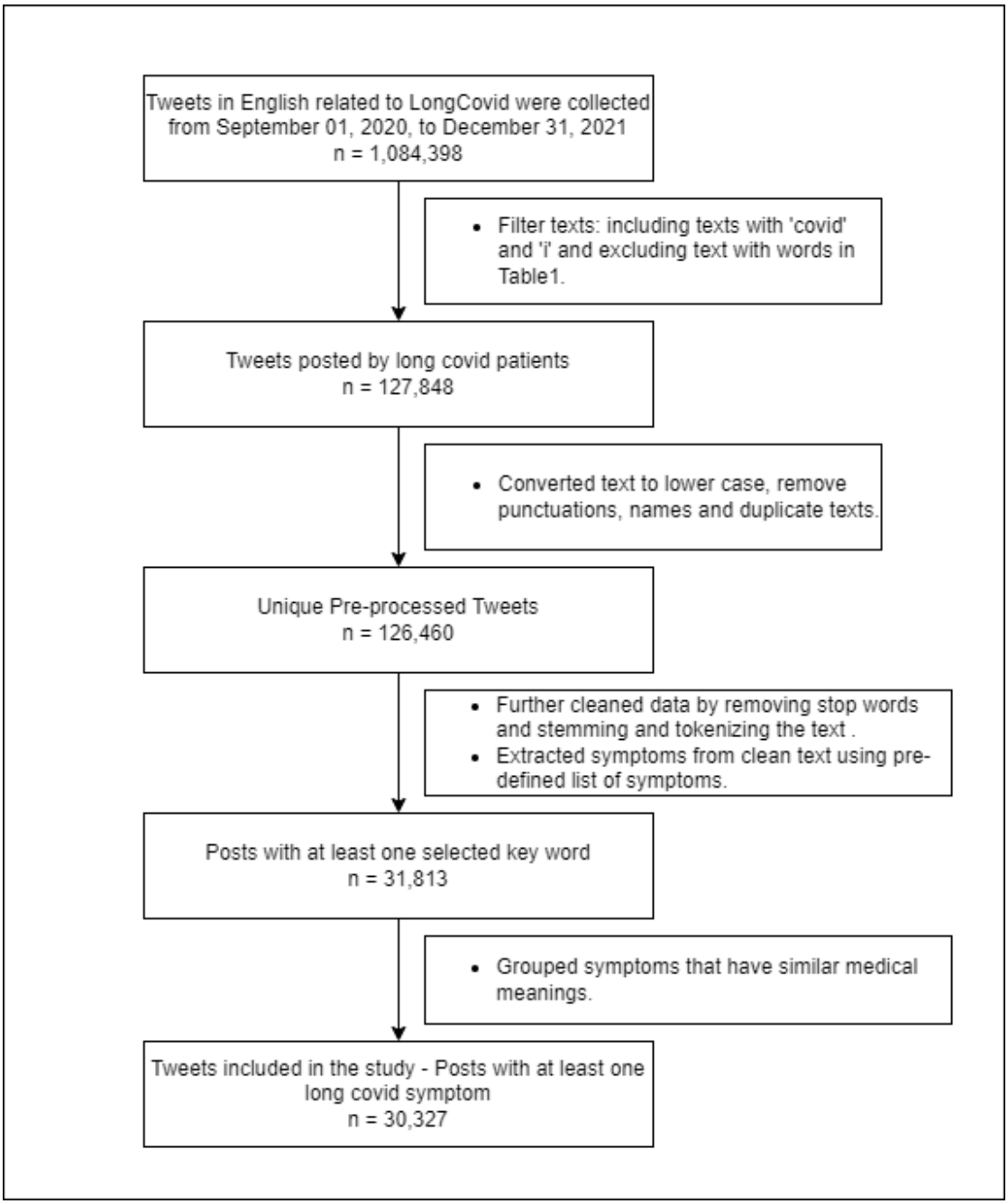


Figure 3.2: Data collection and pre-processing process.

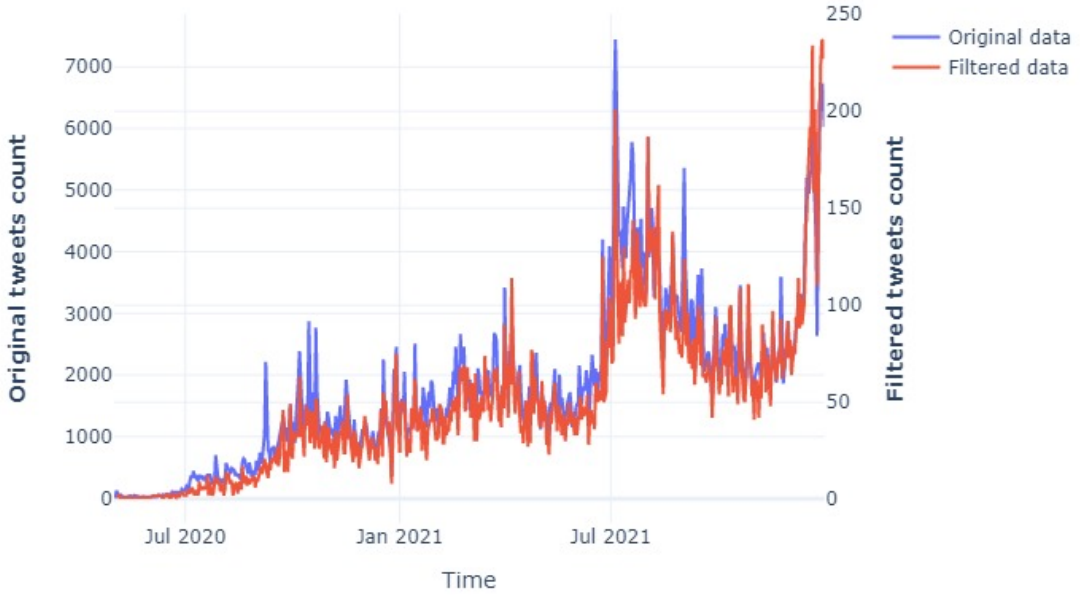


Figure 3.3: Time series plot for originally obtained data and the data considered for the study.

3.2.3 Sentiment Analysis

To measure the sentiment expressed via Twitter on long COVID, we used Sentiment Analysis, a specific type of Natural Language Processing (NLP), computational linguistics, and text analysis (Argamon and Lee; 2009; Liu and Liu; 2012). The subjective information from tweets is analyzed and extracted to classify the text into three classes, namely positive, negative, and neutral. If the polarity is > 0, categorize the text as 'positive,' and if the polarity is < 0, as 'negative.' Texts with 0 polarity values were classified as 'neutral.' The pie charts in Figure 3.4a and Figure 3.4b show the categorization of the sentiment of each text related to long COVID in two stages with all the posts and with posts with at least one long COVID symptoms respectively. The positive sentiment was reduced from 53.1% to 49.8% when we only considered the tweets with symptoms.

Here, we calculated the sentiment polarity of each cleaned and pre-processed tweet using the 'TextBlob' library, which uses the Naive Bayes algorithm for sentiment analysis. TextBlob is a Python library that supports complex analysis and operations on textual data. This is built on the NLTK library (Bird and Klein; 2009) and provides a simple

API for performing several NLP tasks such as sentiment analysis, part-of-speech tagging, noun phrase extraction, classification, and translation.

3.2.4 Collocations

We also understand that symptoms often appear as more than one word in texts. Therefore, finding meaningful symptoms with only two words was a particular task in this study. Many valuable text analyses are based on the relationships between words, examining which words tend to follow others immediately or co-occur. Therefore, we analyzed the relationship between two words of each bigram in tweets and identified which long COVID symptoms appear as a combination of words. We used the NLTK library in python (Bird and Klein; 2009) to identify bi-words from the texts. We filtered them using the pre-prepared list of symptoms (Fig. 7) to get only bi-words that can be meaningful symptoms of long COVID when only appearing together.

To identify the meaningful bi-words, we used the collocation feature in words that reveal a phrase consisting of more than one word. Still, these words commonly co-occur in a given context than their individual word parts. We used several bigram-association measures (Andersen; 2012) to filter out the most meaningful collocations.

1. Pointwise Mutual Information (PMI)

   The Pointwise Mutual Information (PMI) score for two words, w^1 and w^2, is:

   $$PMI(w^1, w^2) = log_2 \frac{P(w^1, w^2)}{P(w^1)P(w^2)}$$

   Here,

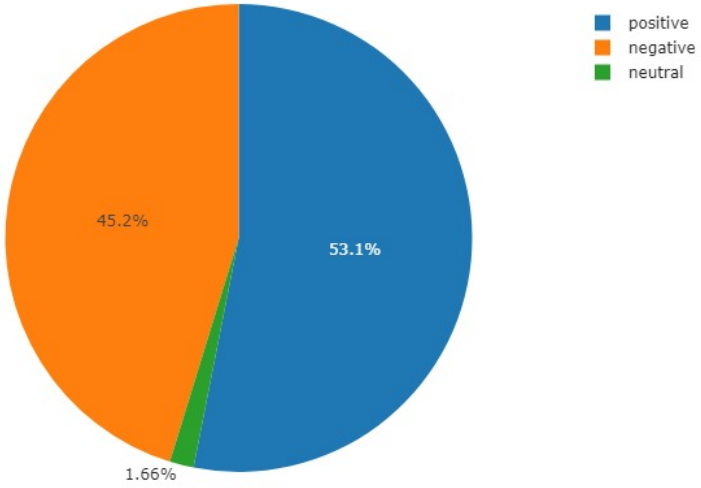
   $$P(w) = \frac{C(w)}{N}$$

   Where C represents the count of each word, and N is the total number of words in the corpus.

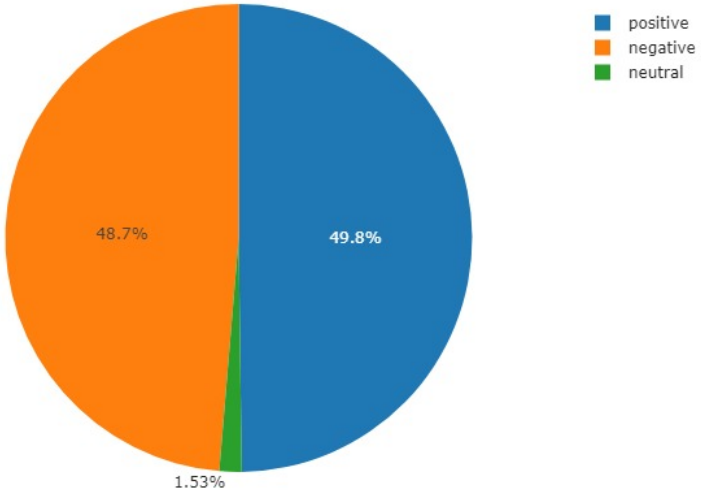
   The main intuition is that it measures how much more likely the words are to co-occur than to occur independently. However, the main disadvantage of this method is that it is very sensitive to a rare combination of words. We handled this issue by using the frequency filter of words.

2. t-test with a frequency filter

   When conducting a t-test, we can consider testing the below hypothesis with a 5% level of significance.



(a) Classification of the sentiment score of all the posts.



(b) Classification of the sentiment score of the posts with at least one long COVID symptom.

Table 3.4: Bigram contingency table for a bigram (xy): observed frequencies (O_{ij}). z_* indicates words that are not x nor y. [a]

	$w^1 = x$	$w^1 \not\equiv x$	
$w^2 = y$	$O_{11} = C(xy)$	$O_{21} = C(zy)$	R_1
$w^2 \not\equiv y$	$O_{21} = C(xz)$	$O_{22} = C(z_1 z_2)$	R_2
	C_1	C_2	N

[a] Here, $C(w^i w^j)$ represents the bi-word count where w^i and w^j appear together. R_i and C_j are the row and column totals, respectively. N indicates the total number of bi-words in the texts.

H_0 : Words (w^1, w^2) occur with probability $\mu : \mu = P(w^1)P(w^2) = \frac{C(w^1)}{N}\frac{C(w^2)}{N}$ versus
H_1 : Words (w^1, w^2) do not occur with probability μ.

The test statistic is:

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{S^2}{n}}}$$

where, $\bar{x} = \frac{c(w^1, w^2)}{n}$ for Bernoulli trial: $S^2 = p(1-p)$, and c represents the count of each word, and n is the total number of words in the sample corpus.

We also avoided the sensitivity to rare cases by filtering the bi-words according to the frequencies. Normality assumption is one of the disadvantages of using the t-test.

3. Chi-square test

The null hypothesis of the chi-square test assumes that words (w^1, w^2) are independent, just like in the t-test. The chi-square test statistic is computed as:

$$\chi^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

The observed frequencies (O_{ij}) and expected frequencies (E_{ij}) can be calculated using the bigram contingency tables presented in Table 3.4 and Table 3.5.

Table 3.5: Bigram contingency table for a bigram (xy): estimated frequencies (E_{ij}). [b]

	$w^1 = x$	$w^1! = x$
$w^2 = y$	$E_{11} = \frac{R_1 C_1}{N}$	$E_{21} = \frac{R_1 C_2}{N}$
$w^2! = y$	$E_{21} = \frac{R_2 C_1}{N}$	$E_{22} = \frac{R_2 C_2}{N}$

[b] R_i and C_j are the row and column totals, respectively. N indicates the total number of bi-words in the texts.

3.2.5 Association Rule Mining (ARM)

In many applications, implications between different situations naturally arise. We call these implications associations. These associations can be discovered and quantified using relational knowledge. "Relational knowledge identifies how concepts/entities are related and how concepts and their relations are defined or described by models" (Halford et al.; 2010). For example, in the Twitter symptom analysis, some symptom rules can be determined, such as "2.7% of long COVID patients who suffer from loss of taste suffer from loss of smell too" or "patients with lung/breathing problems and loss of taste are likely to suffer from loss of smell with 77% confidence". These are called association rules, and the correlation analysis is known as association mining.

ARM (Agrawal et al.; 1993) has become an active research field in the data mining community that can solve various problems in healthcare. Recently, different incremental algorithms have been proposed for mining association rules to discover hidden relationships between symptoms and diseases. This analysis used ARM to obtain insights into the poorly understood LCS by demonstrating the relationships and patterns among the symptoms described in the tweets. We addressed the problem of automatically identifying new and useful symptom patterns in Long COVID patient data using an apriori rule-based data mining algorithm (Argamon and Lee; 2009). Recently, other incremental rule-based algorithms such as Eclat (Zaki et al.; 1997), McEclat (Schlegel et al.; 2013), DHP (Park et al.; 1997), AprioriTID (Li et al.; 2005), and MsApriori (LEE et al.; 2005), have also been introduced for mining association rules to discover hidden relationships between item sets and to enhance the collection of rare frequent events. We used apriori in this paper as it is a well-understood and well-used algorithm out of this category. Computations and related experiments were done using Python.

We can help define LCS for future research and patient care by describing these relationships among symptoms. These patterns expose the combination of the co-occurring symptoms, as it is helpful to know how one symptom/ set of symptoms is associated with others. An association rule between a set of symptoms X and a set of symptoms Y is expressed in the form $X \rightarrow Y$. It is interpreted as 'patients with symptoms X are likely to have symptoms Y.' Generally, the effectiveness of discovered rules is measured in terms of support, confidence, and lift.

Support: Support indicates how frequently the item set appears in the data set.

$$supp(X \rightarrow Y) = supp(Y \rightarrow X) = P(X \cap Y)$$

Confidence: Confidence is the percentage of all transactions satisfying X that satisfy Y. In this analysis, a transaction refers to a single patient's record in the dataset, where each transaction includes the symptoms observed in that patient.

$$conf(X \rightarrow Y) = P(Y|X) = \frac{P(X \cap Y)}{P(X)} = \frac{supp(X \rightarrow Y)}{supp(X)}$$

Lift: If the lift is greater than 1, that lets us know the degree to which those two occurrences depend on one another and makes those rules potentially helpful in predicting the consequences in future data sets.

$$lift(X \rightarrow Y) = lift(Y \rightarrow X) = \frac{P(X \cap Y)}{P(X)P(Y)} = \frac{conf(X \rightarrow Y)}{supp(Y)} = \frac{conf(Y \rightarrow X)}{supp(X)}$$

Thus, based on the analysis of long COVID symptoms, we can mine the association rules among symptoms and quantify their characteristics, such as confidence, support, and lift.

3.3 Statistical Analysis and Results

Information was extracted for 1,084,398 individual tweets, of which 34,022 had reported long COVID symptoms (Figure 3.2). The first five results obtained from each collocation method discussed in the previous section are shown in Figure 3.5. After observing the results, we determined that the t-test with a filter method achieves acceptable results, as confirmed by medical experts. This means that the method effectively identified symptom collocations that align with clinically recognized patterns of long COVID, ensuring that the associations found were both medically relevant and meaningful.

We also had to manually select the meaningful bi-words with a test statistic > 5 and have a unique meaning only when they are together. We identified 20 such stemmed bi-words which appeared together frequently, namely, 'brain fog,' 'loss taste,' 'loss smell,' 'chest pain,' 'cant walk,' 'barely walk,' 'unable walk,' 'struggle walk,' 'couldnt walk,' 'stair walk,' 'joint pain,' 'muscle pain,' 'leg pain,' 'lose weight,' 'loss weight,' 'gain weight,' 'cant sleep,' 'sore throat,' 'pin & needle' and 'heart palpitation.'

	PMI	T-test with Filter	Chi-Sq Test
0	(runni, nose)	(brain, fog)	(brain, fog)
1	(pin, needl)	(mental, health)	(glandular, fever)
2	(shortterm, memori)	(chronic, fatigu)	(runni, nose)
3	(glandular, fever)	(tast, smell)	(mental, health)
4	(hay, fever)	(viral, fatigu)	(sore, throat)

Figure 3.5: First five bi-words selection from each method.

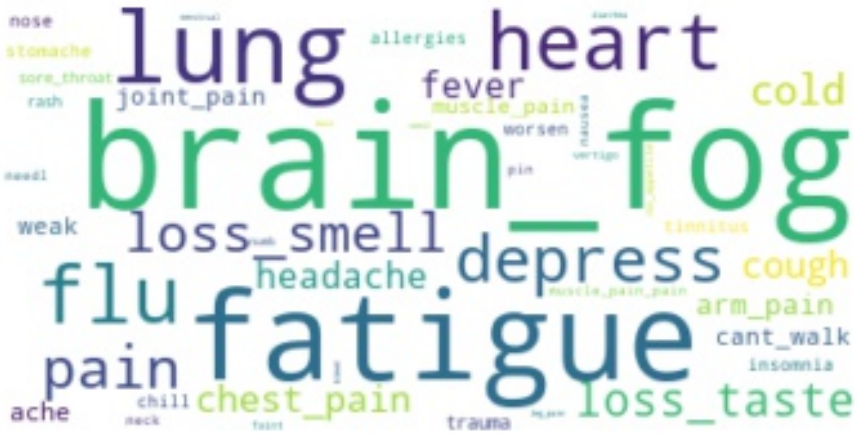


Figure 3.6: Word cloud of the list of 44 identified symptoms.

Because some bi-words have a similar medical meaning, we grouped similar words into categories for analysis (Table 3.3). After grouping symptoms, the total number of symptoms to be analyzed was decreased from 73 to 44. Figure 3.6 shows the reduced set of symptoms, and the size parameter of the word cloud indicates the frequency of each symptom appearing in the tweets. Brain fog (26%) was the most common symptom, followed by fatigue (17%), breathing/lung issues (16%), heart issues (10%), flu symptoms (9%), depression (7%), and pain site of which is not explicitly mentioned (6%). Loss of smell and taste, cold, cough, chest pain, fever, headache, and arm pain were each reported in 2–5% of tweets. Symptoms such as aches, weakness, joint pain, walking difficulties, muscle pain, trauma, allergies, worsened symptoms, tinnitus, insomnia, nose-related issues, stomach, and chills were reported in 1% of the tweets, while the rest of the symptoms were in less than 1% of tweets (Figure 3.7).

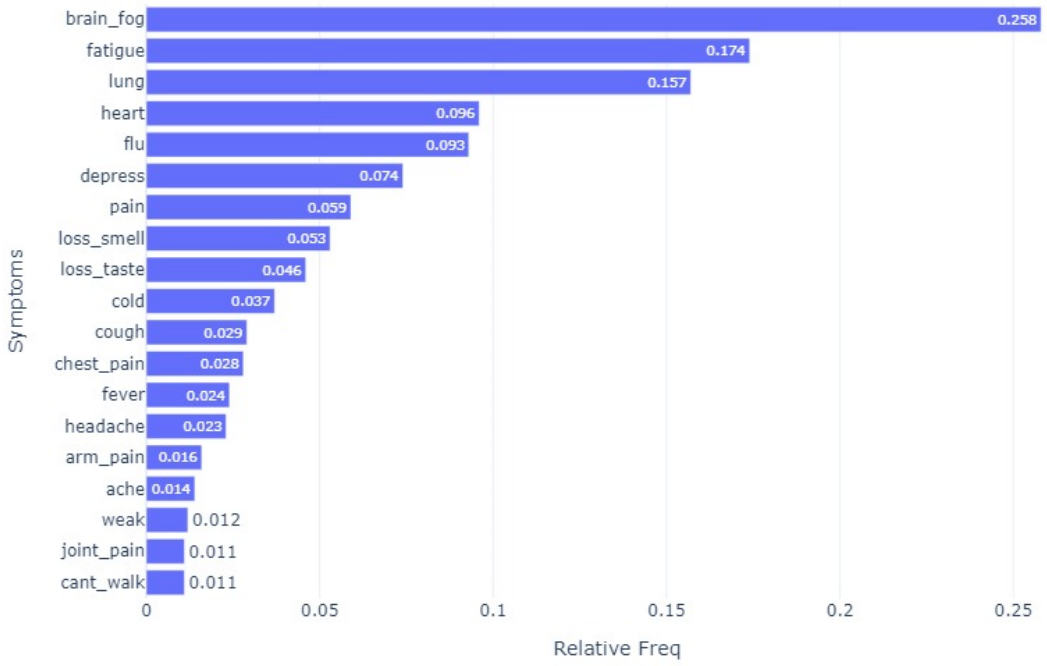


Figure 3.7: Relative frequency of symptoms in long COVID patients.

Figure 3.8 presents how each symptom appears in discussions over time, and the pattern is similar to the number of tweets shown in Figure 3.3. Hence, there is no noticeable difference in symptoms with time.

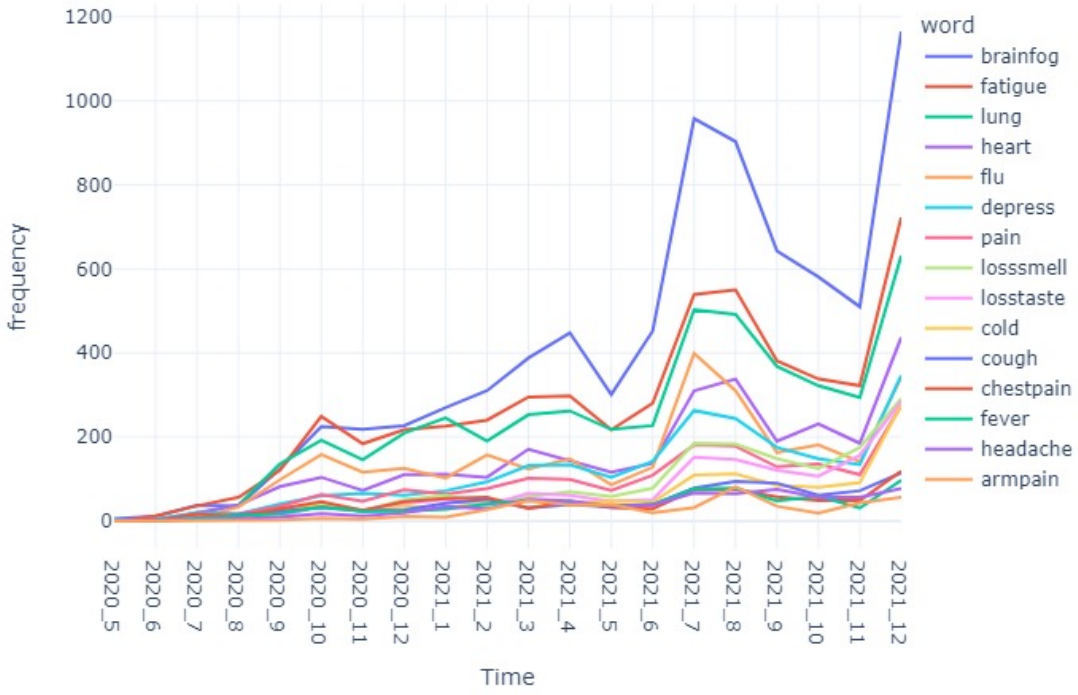


Figure 3.8: Most frequent 15 long COVID symptoms by time.

3.3.1 Symptom Rules

Our study considered each tweet as a single transaction coming from a single individual. We applied ARM to the symptom data considering one tweet as one transaction and identified symptom rules. The ARM algorithm using symptom transactions aims to construct frequent item sets, having at least a user-specified threshold. Thus, we set a "confidence" threshold of 0.1 or 10%. We set up a minimum support threshold value above 0.001 and a "lift" greater than 1 for positively correlated rules. We discovered 57 significant rules for the data that included symptom-only information and presented them in Table 3.6.

The highest confidence level-based detection was used to determine the most significant rules. Among the top 12 rules that have confidence > 0.3, loss of smell and loss of taste were the most common consequent symptoms, followed by lung/breathing problems and fatigue. If a patient had lung/breathing problems and loss of taste, there was a 77% confidence that they had the loss of smell. Similarly, patients with fatigue and loss of taste also had a loss of smell as a consequent symptom. The top 12 rules are visualized in Figure 3.9. We describe rule number eleven, represented by R11 in a yellow node.

There are three symptom nodes, “fatigue,” “cough,” and “lung,” represented by green nodes. These three nodes form a rule where the antecedents are “fatigue,” “cough,” and the consequent is “lung.” Both the nodes in the antecedent have outgoing links, which are pointed toward the R11 node. Similarly, there is an outgoing link in R11, which is pointing toward the consequent “lung” node.

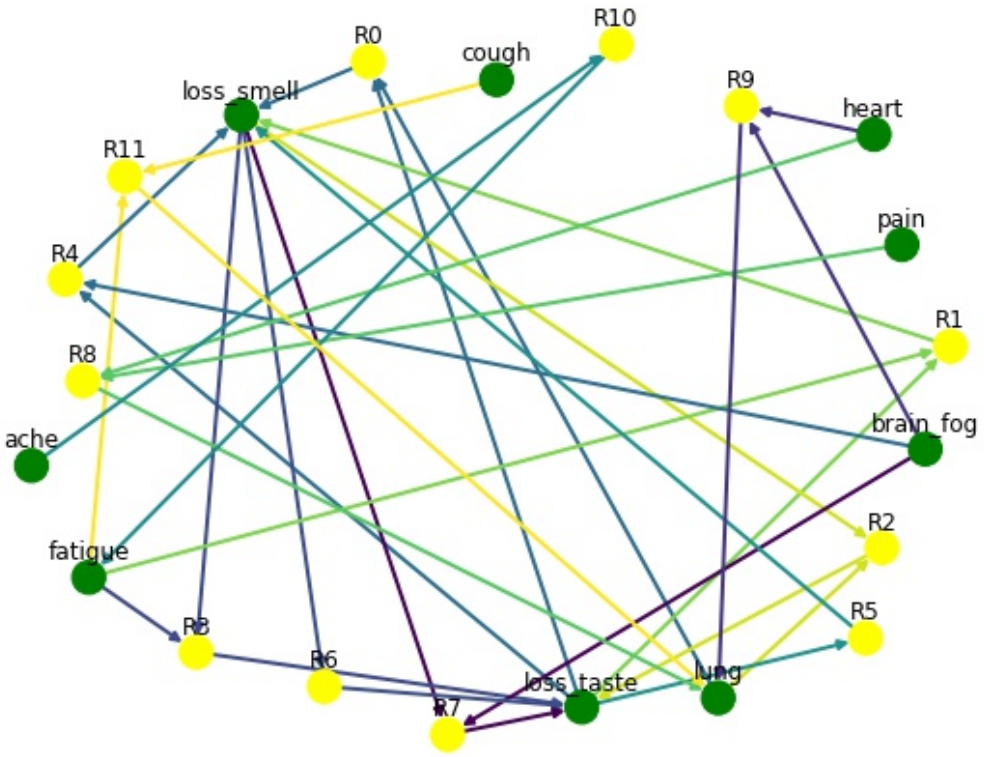


Figure 3.9: Association rules visualization.

Table 3.6: List of identified symptom rules.

Rule	Antecedents	Consequents	Support	Confidence	Lift
R0	(loss_taste, lung)	(loss_smell)	0.0028	0.7748	14.54
R1	(loss_taste, fatigue)	(loss_smell)	0.0028	0.7368	13.8281
R2	(loss_smell, lung)	(loss_taste)	0.0028	0.7107	15.3196
R3	(loss_smell, fatigue)	(loss_taste)	0.0028	0.6614	14.2564
R4	(loss_taste, brain_fog)	(loss_smell)	0.0018	0.5978	11.2192
R5	(loss_taste)	(loss_smell)	0.0272	0.5871	11.0173
R6	(loss_smell)	(loss_taste)	0.0272	0.5111	11.0173
Continued on next page					

Table 3.6 – continued from previous page

Rule	Antecedents	Consequents	Support	Confidence	Lift
R7	(loss_smell, brain_fog)	(loss_taste)	0.0018	0.4911	10.5847
R8	(heart, pain)	(lung)	0.0012	0.3684	2.3522
R9	(heart, brain_fog)	(lung)	0.0032	0.3542	2.2617
R10	(ache)	(fatigue)	0.0047	0.3471	1.9921
R11	(fatigue, cough)	(lung)	0.0012	0.3083	1.9686
R12	(heart, fatigue)	(lung)	0.0021	0.3014	1.9246
R13	(headache)	(fatigue)	0.0062	0.2675	1.5354
R14	(brain_fog, lung)	(heart)	0.0032	0.2574	2.6915
R15	(heart)	(lung)	0.0223	0.2328	1.4861
R16	(lung, pain)	(fatigue)	0.0016	0.2308	1.3245
R17	(insomnia)	(fatigue)	0.0013	0.2229	1.2791
R18	(muscle_pain)	(fatigue)	0.0019	0.2159	1.2392
R19	(muscle_pain)	(ache)	0.0019	0.2159	15.8929
R20	(cough)	(lung)	0.0061	0.2126	1.3572
R21	(lung, cough)	(fatigue)	0.0012	0.1989	1.1417
R22	(chest_pain)	(lung)	0.0053	0.1891	1.2075
R23	(fever)	(fatigue)	0.0046	0.1871	1.0737
R24	(pain)	(fatigue)	0.0102	0.1736	0.9962
R25	(cold)	(flu)	0.0064	0.1721	1.8483
R26	(lung, pain)	(heart)	0.0012	0.1683	1.7597
R27	(muscle_pain)	(lung)	0.0015	0.1667	1.0641
R28	(weak)	(fatigue)	0.0019	0.1633	0.9374
R29	(trauma)	(brain_fog)	0.0012	0.1586	0.6157
R30	(ache)	(lung)	0.0021	0.1553	0.9918
R31	(fatigue, pain)	(lung)	0.0016	0.1548	0.9886
R32	(ache)	(pain)	0.002	0.1505	2.5553
R33	(heart, lung)	(brain_fog)	0.0032	0.1422	0.5521
R34	(lung)	(heart)	0.0223	0.1421	1.4861
R35	(brain_fog, lung)	(fatigue)	0.0017	0.1421	0.8155
R36	(joint_pain)	(brain_fog)	0.0016	0.142	0.5514
R37	(ache)	(headache)	0.0019	0.1383	6.0025
R38	(ache)	(muscle_pain)	0.0019	0.1383	15.8929
R39	(cough)	(fatigue)	0.004	0.1371	0.7871
R40	(arm_pain)	(fatigue)	0.0021	0.135	0.7749

Continued on next page

Table 3.6 – continued from previous page

Rule	Antecedents	Consequents	Support	Confidence	Lift
R41	(muscle_pain)	(pain)	0.0012	0.1326	2.2512
R42	(brain_fog, fatigue)	(lung)	0.0017	0.1274	0.8134
R43	(muscle_pain)	(weak)	0.0011	0.1212	10.533
R44	(weak)	(lung)	0.0014	0.1203	0.7684
R45	(fatigue, lung)	(heart)	0.0021	0.1198	1.2525
R46	(pain)	(lung)	0.0069	0.1165	0.7436
R47	(ache)	(fever)	0.0015	0.1141	4.6563
R48	(joint_pain)	(pain)	0.0013	0.113	1.9195
R49	(lung)	(fatigue)	0.0173	0.1107	0.6356
R50	(depress)	(brain_fog)	0.0081	0.1095	0.425
R51	(headache)	(lung)	0.0025	0.1087	0.6942
R52	(loss_smell, loss_taste)	(lung)	0.0028	0.1041	0.6647
R53	(depress)	(fatigue)	0.0076	0.1024	0.5877
R54	(loss_smell, loss_taste)	(fatigue)	0.0028	0.1017	0.5837
R55	(joint_pain)	(ache)	0.0012	0.1014	7.4676

3.4 Discussion

3.4.1 Principal Findings

The symptoms associated with LCS are still poorly understood. Analyzing social media conversations of patients related to long COVID allows us to understand the frequency and relationship between symptoms. Based on a large amount of Twitter social media data related to LCS, we performed the following two tasks in this paper. First, we identified the symptoms and medical conditions related to long COVID that were discussed on the Twitter social media platform. Second, we determined the patterns of symptoms and their associations.

Brain fog, fatigue, and breathing/lung issues were the three most common symptoms identified by the analysis. The literature sources verified these reported symptoms (Editorial; 2020; Soriano et al.; 2022). With a strong relationship, loss of taste and loss of smell were the most common consequent symptoms. We also identified a substantial number of meaningful symptom rules for a predefined threshold for confidence and support. Only the positively correlated relationships were considered by setting the lift to

be greater than 1. The number of strong association rules can be changed according to different minimum thresholds. By finding associations among long COVID symptoms, this work will help physicians identify the behavior of the patients with long COVID. This paper provides new ideas for symptom mining and reveals the relationship between symptoms and their application value. Thus, the work has theoretical and practical importance.

We have used a novel data source, Twitter, and multiple NLP and machine learning techniques to explore the symptoms described by a large undifferentiated population. Different NLP methods, such as sentiment analysis, keyword extraction, and lemmatization, were used to extract information and symptoms from the unstructured text data. Subsequently, we used ARM concepts (Agrawal et al.; 1993) to expose meaningful relationships between long COVID symptoms. This proof of concept has demonstrated the value of these techniques in describing the frequency of symptoms and the relationships between more common symptoms.

Future research can build on this methodology with clinical data sources such as electronic medical records, adding individual covariates such as sex, age, location, and comorbidities. This research can also be further extended to detect and predict the consequences of a given set of symptoms using word popularity detection methods.

3.4.2 Limitations

This study is based on web-based Twitter data with limited patient-level variables. No information about the demographics of the tweet authors was available. Furthermore, we only considered patients who shared their experiences with the public in English, as including English-language tweets and, therefore, translating them is computationally intensive. Another limitation of the study is that results can be affected by misinformation or false conversations on the Twitter platform.

There is a caveat for the confidence metric in the ARM technique when a negative correlation exists between the two sets, for instance, $\sim X \rightarrow Y$. In most cases, lower support and confidence are preferred when examining negatively correlated rules. The positive symptoms are often clear; however, negative symptoms are subtler and more difficult to recognize and diagnose. As a result, we have not investigated negatively correlated rules. Furthermore, rules discovered by algorithms require clinical validation and verification.

3.4.3 Conclusion

The most frequent symptoms in our study included brain fog, fatigue, breathing/lung issues, heart issues, flu symptoms, and depression. General pains, loss of smell and taste, cold, cough, chest pain, fever, headache, and arm pain emerged in 1.6% to 5.3% of patients with long COVID. Furthermore, the highest confidence level–based detection with 10% minimum confidence and 0.01% minimum support successfully demonstrates the potential of association analysis and the apriori algorithm to establish patterns to explore 57 meaningful relationship rules among long COVID symptoms. In this study, to identify the positively correlated symptoms, we only considered the rules with the lift being greater than 1. The results revealed that patients with lung/breathing problems and loss of taste are likely to have loss of smell with 77% confidence.

3.4.4 Acknowledgments

This study was funded through a Canadian Institutes of Health Research Operating Grant: Emerging COVID-19 Research Gaps and Priorities Funding Opportunity.

3.4.5 Conflicts of Interest

None declared.

Bibliography

Agrawal, R., Imieliński, T. and Swami, A. (1993). Mining association rules between sets of items in large databases, *SIGMOD record* **22**(2): 207–216.

Allen, J. (1995). *Natural language understanding*, [2nd ed.]. edn, Benjamin/Cummings Pub. Co., Redwood City, Calif.

Andersen, G. (2012). *Exploring Newspaper Language: Using the Web to Create and Investigate a Large Corpus of Modern Norwegian*, Studies in Corpus Linguistics, 1 edn, John Benjamins Publishing Company, Philadelphia.

Argamon, Shlomo (Review of: Pang, B. and Lee, L. (2009). Opinion mining and sentiment analysis, *Computational Linguistics* **35**(2): 311–312.

Bird, Steven, E. L. and Klein, E. (2009). *Natural Language Processing with Python*, O'Reilly Media Inc.

Callard, F. and Perego, E. (2021). How and why patients made long covid, *Social science & medicine (1982)* **268**: 113426–113426.

Chen, X., Xie, H., Wang, F. L., Liu, Z., Xu, J. and Hao, T. (2018). A bibliometric analysis of natural language processing in medical research, *BMC medical informatics and decision making* **18**(Suppl 1): 14–14.

Chowdhary, K. (2020). *Fundamentals of Artificial Intelligence*, 1st edition edn, Springer Nature, Netherlands.

Editorial, N. (2020). Long covid: let patients help define long-lasting covid symptoms, *Nature (London)* **586**(7828): 170–170.

Fitzgerald, R. T. and Radmanesh, A. (2015). Social media and research visibility, *American journal of neuroradiology : AJNR* **36**(4): 637–637.

for Disease Control, C. and Prevention (2022). Long covid or post-covid conditions. `https://www.cdc.gov/coronavirus/2019-ncov/long-term-effects/index.html`[Accessed: 2022-03-05].

Halford, G. S., Wilson, W. H. and Phillips, S. (2010). Relational knowledge: the foundation of higher cognition, *Trends in cognitive sciences* **14**(11): 497–505.

Joshee, S., Vatti, N. and Chang, C. (2022). Covid-19: Long-term effects, *Mayo Clinic proceedings* **97**(3): 579–599.

LEE, Y.-C., HONG, T.-P. and LIN, W.-Y. (2005). Mining association rules with multiple minimum supports using maximum constraints: Data mining and granular computing, *International journal of approximate reasoning* **40**(1-2): 44–54.

Li, Z.-C., He, P.-L. and Lei, M. (2005). A high efficient aprioritid algorithm for mining association rule, *2005 International Conference on Machine Learning and Cybernetics*, Vol. 3, IEEE, pp. 1812–1815 Vol. 3.

Liu, B. and Liu, B. (2012). *Sentiment analysis and opinion mining*, Vol. 16 of *Synthesis lectures on human language technologies*, Morgan & Claypool Publishers.

NHS (2022). Long-term effects of COVID-19. `https://www.nhsinform.scot/long-term-effects-of-covid-19-long-covid`[Accessed: 2022-03-05].

Park, J., Yu, P. and Chen, M.-S. (1997). Mining association rules with adjustable accuracy, *Conference on Information and Knowledge Management: Proceedings of*

the sixth international conference on Information and knowledge management; 10-14 Nov. 1997, ACM, pp. 151–160.

Reddy, S. C., Kathiravan, S. and Singh, S. M. (2023). An analysis of self-reported long covid-19 symptoms on twitter, *Journal of Postgraduate Medicine, Education and Research* **57**(2): 79–81.

Schlegel, B., Karnagel, T., Kiefer, T. and Lehner, W. (2013). Scalable frequent itemset mining on many-core processors, *Proceedings of the Ninth International Workshop on data management on new hardware*, ACM, pp. 1–8.

Siddiqui, S. and Singh, T. (2016). Social media its impact with positive and negative aspects, *International Journal of Computer Applications Technology and Research* **5**(2): 71–75.

Soriano, J. B., Murthy, S., Marshall, J. C., Relan, P. and Diaz, J. V. (2022). A clinical case definition of post-covid-19 condition by a delphi consensus, *The Lancet infectious diseases* **22**(4): e102–e107.

Stilou, S., Bamidis, P., Maglaveras, N. and Pappas, C. (2001). Mining association rules from clinical databases: an intelligent diagnostic process in healthcare., *Stud Health Technol Inform.* **84**(2).

Yong, E. (2022). Covid-19 can last for several months. `https://www.theatlantic.com/health/archive/2020/06/covid-19-coronavirus-longterm-symptoms-months/612679/`[Accessed: 2022-03-05].

Zaki, M. J., Parthasarathy, S., Ogihara, M. and Li, W. (1997). Parallel algorithms for discovery of association rules, *Data mining and knowledge discovery* **1**(4): 343–.

Chapter 4

Long COVID Prediction in Manitoba Using Clinical Notes Data: A Machine Learning Approach

The content of this chapter is derived from the manuscript text of the under-review publication, cited as follows:

Matharaarachchi, S., Domaratzki, M., Katz, A. and Muthukumarana, S. (2024). Long COVID Prediction in Manitoba Using Clinical Notes Data: A Machine Learning Approach (In Review).

Long COVID Prediction in Manitoba Using Clinical Notes Data: A Machine Learning Approach

Surani Matharaarachchi[a,*], Mike Domaratzki[b], Alan Katz[c], Saman Muthukumarana[a]

[a]*Department of Statistics, University of Manitoba, Winnipeg, MB, Canada*
[b]*Department of Computer Science, Western University, London, ON, Canada*
[c]*Departments of Community Health Sciences and Family Medicine, University of Manitoba, Winnipeg, MB, Canada*

Abstract

Identifying patients afflicted by long COVID Syndrome (LCS) presents a substantial and intricate challenge. Nonetheless, this paper introduces a promising avenue by using the capabilities of machine learning classification models adept at unravelling the intricacies of distinguishing LCS cases from many other health conditions. Using natural language processing techniques, we have outlined a confirmed LCS group to serve as the foundation for classification. This study aims to establish a precise and dependable method for identifying LCS patients, achieved through a comprehensive assessment of classification and re-sampling techniques. This includes the integration of patient attributes and both pre- and post-COVID symptoms linked to LCS as input variables.

To evaluate the performance of the models, we employ key metrics such as sensitivity, specificity, and ROC (Receiver Operating Characteristic) analysis. Our findings emphasize that logistic regression with elastic net regularization combined with Random Under-Sampling is the most effective predictive methodology. This method successfully identified patients at risk of LCS, achieving a sensitivity of 0.95, a specificity of 0.81, and an AUC of 0.94. Notably, it identifies 1124 (24.7%) LCS patients from the set of 4556 COVID-19 cases, more than 14 times the number of known cases. These insights can substantially contribute to developing a consistent and efficient approach for identifying

LCS patients, thereby enhancing our comprehension of the condition and facilitating more precise diagnoses and improved treatment strategies.

Keywords: *Long COVID Syndrome; Machine Learning; Elastic Net Classification; Natural Language Processing; Class Imbalance; Random Under-Sampling*

4.1 Introduction

Post-acute sequelae of SARS-CoV-2 infection (PASC), also referred to as long COVID Syndrome (LCS), is a condition in which individuals experience symptoms for weeks or months after recovering from COVID-19 (Nalbandian et al.; 2021). It is also known as post-COVID-19 condition (PCC) and can affect both adults and children. The condition can affect multiple organ systems, including respiratory, cardiovascular, neurological, and psychological (Carfì et al.; 2020; Huang et al.; 2021), significantly impacting patients' quality of life and ability to perform day-to-day activities.

The need for consistent identification and treatment of LCS patients is a significant problem facing the healthcare community. Additionally, many patients who continue to experience lingering symptoms are either unaware of their condition or lack access to timely healthcare services. According to a large cohort study by Huang et al. (2021), approximately 20%-30% of individuals diagnosed with COVID-19 experience persistent symptoms with longer duration, indicating the need for improved identification and management of LCS patients (Binka et al.; 2022; Bull-Otterson et al.; 2022; *Characterizing Long COVID in an international cohort: 7 months of symptoms and their impact*; 2021; Crook et al.; 2021; Amin-Chowdhury and Ladhani; 2021).

This paper demonstrates the feasibility of using natural language processing and supervised machine learning tools to improve understanding of complex diseases that lack a well-understood definition exclusively using patient data. These advanced approaches aim to make distinguishing LCS cases from other conditions easier. The study compares different models using patient attributes and symptoms as inputs to find an accurate and reliable method for identifying LCS patients. This research could significantly improve the accuracy of diagnosing LCS and enhance patient care.

4.1.1 Related Work

Patient data can be collected from electronic health records, patient surveys, or other sources to train machine-learning classification models for LCS. The data can include

patient attributes such as age, sex, socioeconomic indicators, and symptoms experienced before and following COVID-19. Symptoms such as fatigue, dyspnea, and joint pain have been shown to be associated with LCS (Sudre et al.; 2021).

Recent studies have shown that specific patient attributes and symptoms are associated with an increased risk of LCS (Blomberg et al.; 2021; Sudre et al.; 2021; Binka et al.; 2022). These findings have led researchers to investigate using machine learning techniques to identify patients at risk for LCS and associated risk factors. Machine learning techniques, such as elastic net classification and random forest, have been explored to predict the risk of LCS by training algorithms with patient information and symptoms.

A study by Sudre et al. (2021) developed a machine learning model using a random forest classifier to predict the risk of LCS using patient data from the UK Biobank. Data for this analysis were collected from a mobile health application. The model achieved an area under the receiver operator characteristic curve (AUC) of 0.76, indicating a moderate ability to discriminate between those with and without LCS. The study found that the most important predictors of LCS were fatigue, headache, and dyspnea. In addition, the study also found that a previous diagnosis of anxiety or depression was associated with an increased risk of LCS. Pfaff et al. (2022) developed XGBoost machine learning models to identify potential patients with LCS, achieving AUC of 0.92 (all patients), 0.90 (hospitalized), and 0.85 (non-hospitalized). According to the Shapley values, significant features of their analysis include the rate of healthcare utilization, patient age, dyspnea, and additional diagnosis and medication information accessible in the electronic health record. With an AUC of 0.93, Binka et al. (2022) have identified 18% of COVID-positive people as LCS using elastic net regression models. Two latter studies used LCS clinic patient visits to confirm the known LCS group.

Using multivariable logistic regression models, Margalit et al. (2022) assessed risk factors for significant LCS fatigue and concluded that it is related to autonomic dysfunction, impaired cognition, and decreased mood. Factor analysis has also been applied to identify six different patterns of the 14 symptoms, where the cognitive pattern was the major symptom pattern, explaining 26.2% of the variance (Yelin et al.; 2022).

These studies highlight the potential of machine learning models for the early identification and management of LCS. However, further research is needed to validate and refine these models before they can be widely used in clinical practice.

By combining Natural Language Processing (NLP) tools with binary classification while addressing the class imbalance, our study aims to achieve a comprehensive approach to predicting at-risk LCS patients. NLP is used to identify symptoms and confirmed LCS cases, and binary classification is applied to distinguish these cases from

patients with no LCS group. This integrated approach facilitates early diagnosis and effective management of LCS.

4.2 Methods and Experimental Design

4.2.1 Data Collection

This study used data from the Manitoba COVID-19 Cohort, encompassing information on all adult individuals who underwent testing for severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and reported cases of COVID-19 in Manitoba between March 1, 2020, and December 31, 2021. A sample of the COVID-19 test-positive cohort, including their demographic information such as age, sex, and socioeconomic factor index (SEFI) (Chateau et al.; 2012), was accessed from the Manitoba Population Research Data Repository housed at the Manitoba Centre for Health Policy (MCHP) (*A systematic investigation of Manitoba's Provincial Laboratory data*; 2012). The SEFI, derived from Census data, indicates non-medical social determinants of health and functions as a proxy for socioeconomic status (SES). The SEFI, to determine an aggregate score, considers four key variables: the average household income, the percentage of single-parent households, the unemployment rate among individuals aged 15 and older, and the high school education rate.

This information was integrated with electronic medical records (EMRs), including doctors' encounter notes, from the Manitoba Primary Care Research Network (MaPCReN) to extract relevant LCS symptoms. MaPCReN is the Manitoba subnetwork of the Canadian Primary Care Sentinel Surveillance Network (CPCSSN) (Birtwhistle; 2011), and we received encounter notes from March 1, 2020 to September 28, 2022. However, Interlake-Eastern Regional Health Authority joined MaPCReN in 2021, and Northern RHA (NRHA) has not provided data since 2020. The largest region (WRHA), which includes 57% of the population (as of 2021), does not permit sharing patient encounter notes from its direct managed clinics. Table 4.1 describes the limitation of the representativeness of the sample. The percentage of the population for each RHA in 2021 is as reported in the Manitoba Health Population Report - June 1, 2021.

Table 4.1: MaPCReN patients with encounter note records in April-June 2022

*RHA	Number of encounter notes	Number of patients with ≥ 1 note	With no encounter note records	Number of patients	Population % for RHA (2021)
IERHA	8,194	3,344	6,017	9,361	9.7%
PMRHA	9,913	2,763	9,122	11,885	12.4%
SRHA	57,891	22,529	56,561	79,090	15.5%
WRHA	72,881	28,954	143,690	172,644	56.9%
NRHA	-	-	-	-	5.5%
Total	148,879	57,590 (21.1%)	215,390 (78.9%)	272,980	100%

* RHA - Regional Health Authority, IERHA - Interlake-Eastern RHA, PMRHA - Prairie Mountain RHA, SRHA - Southern Health-Santé Sud RHA, WRHA - Winnipeg RHA, NRHA - Northern RHA

4.2.2 Study Sample

Eligibility criteria for this analysis were based on patients who had received a COVID-19 index date (the earliest positive test result reported date) from March 1, 2020, to December 31, 2021. After restricting to individuals where EMRs were available, the data set was narrowed to 4556 COVID-19-positive patients with written medical records. The assessment of post-COVID symptoms commenced 90 days after the COVID-19 index date in keeping with the WHO definition of LCS. In contrast, the observation period for assessing pre-COVID symptoms was expanded to encompass two scenarios: one considering symptoms within 90 days before the COVID-19 index date and the other examining symptoms within one year before the COVID-19 index date. This timeline allowed for the evaluation of symptoms before and after the onset of COVID-19, providing a comprehensive understanding of the symptom profile during the specified observation periods.

4.2.3 Identifying the Known LCS Group

The capacity of a computer to comprehend human language, whether spoken or written, is known as Natural Language Processing (NLP) (Celi et al.; 2020; Keselj and Martin; 2009). This manuscript uses a word extraction method to identify EMRs, specifically discussing LCS. As the understanding of LCS evolves, it becomes increasingly important to extract and analyze relevant information from medical records efficiently. Manual review of all the encounter notes can be a time-consuming and labour-intensive process, necessitating the need for automated techniques. The word extraction method combines NLP techniques with domain-specific knowledge to effectively identify encounter notes containing LCS information.

The word extraction method we employed involves a multi-step process. Initially, we compiled a comprehensive list of LCS-related keywords through an extensive literature review and expert input. These keywords encompass various synonyms that explain LCS, such as Long COVID, Post COVID, and Long Hauler. Subsequently, we developed a custom algorithm that utilizes NLP techniques, including tokenization, stemming, and regular expressions, to scan the encounter notes and detect instances of these LCS-related keywords.

In Fig. 4.1, the illustration depicts the refinement process of the established LCS group by using encounter notes. In step 3, we identified 121 patients with probable LCS-related encounter notes. In step 5, an expert review is introduced to validate NLP-filtered encounter notes further. Out of 121 patients identified, 81 were identified as confirmed LCS patients, eight were determined to be unrelated to LCS, and the remaining 32 were categorized as having unknown conditions and subsequently added to the application dataset.

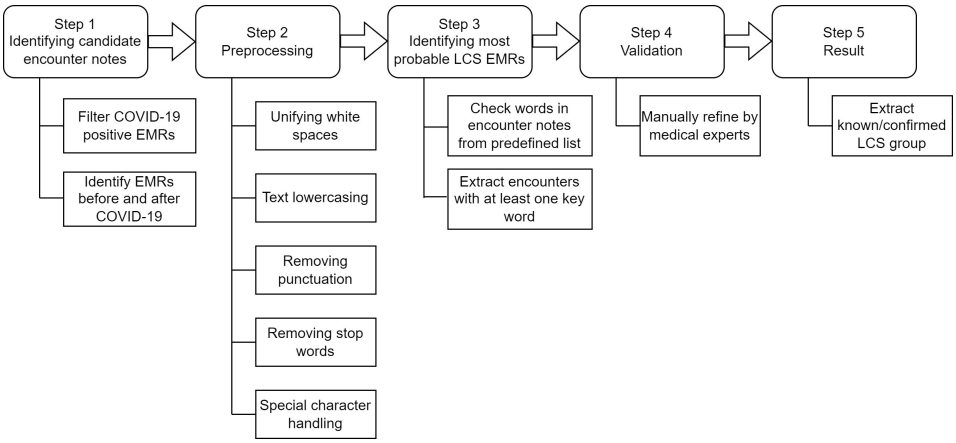


Figure 4.1: Identifying confirmed LCS group. LCS - Long COVID Syndrome, EMRs - Electronic Medical Records.

4.2.4 Identifying the Control Group

According to Amin-Chowdhury and Ladhani (Amin-Chowdhury and Ladhani; 2021), the lack of appropriate controls is a significant limitation for designing machine learning tools, especially in assessing longer-term sequelae. Some studies have included symptomatic but COVID-19 test–negative people as control participants (Binka et al.; 2022). Some jurisdictions have LCS clinics that can help identify training data; for example, in a study with a cohort in British Columbia, the two classes provided to the algorithm were LCS clinic patients and patients who did not attend the LCS clinic (Pfaff et al.;

2022). Some others have compared patients with persistent symptoms with short-lasting illnesses.

In this study, we implemented an approach to identify a control group of patients who did not manifest LCS symptoms. Specifically, we defined the control group as comprising 1945 patients who remained within the dataset for at least 90 days with no documented medical records beyond 90 days from COVID-19 onset (Fig. 4.2). This methodology is consistent with previous investigations examining LCS patients and control groups (Moldofsky and Patcai; 2011). Our rationale for selecting patients without LCS symptoms was to establish a cohort of individuals with a comparable COVID-19 disease trajectory to those without long-term symptoms. This facilitated the isolation of LCS-specific symptoms and patient attributes. Moreover, our selection criteria of patients who remained within the dataset for at least 90 days ensured adequate follow-up data to accurately classify patients as LCS or no LCS.

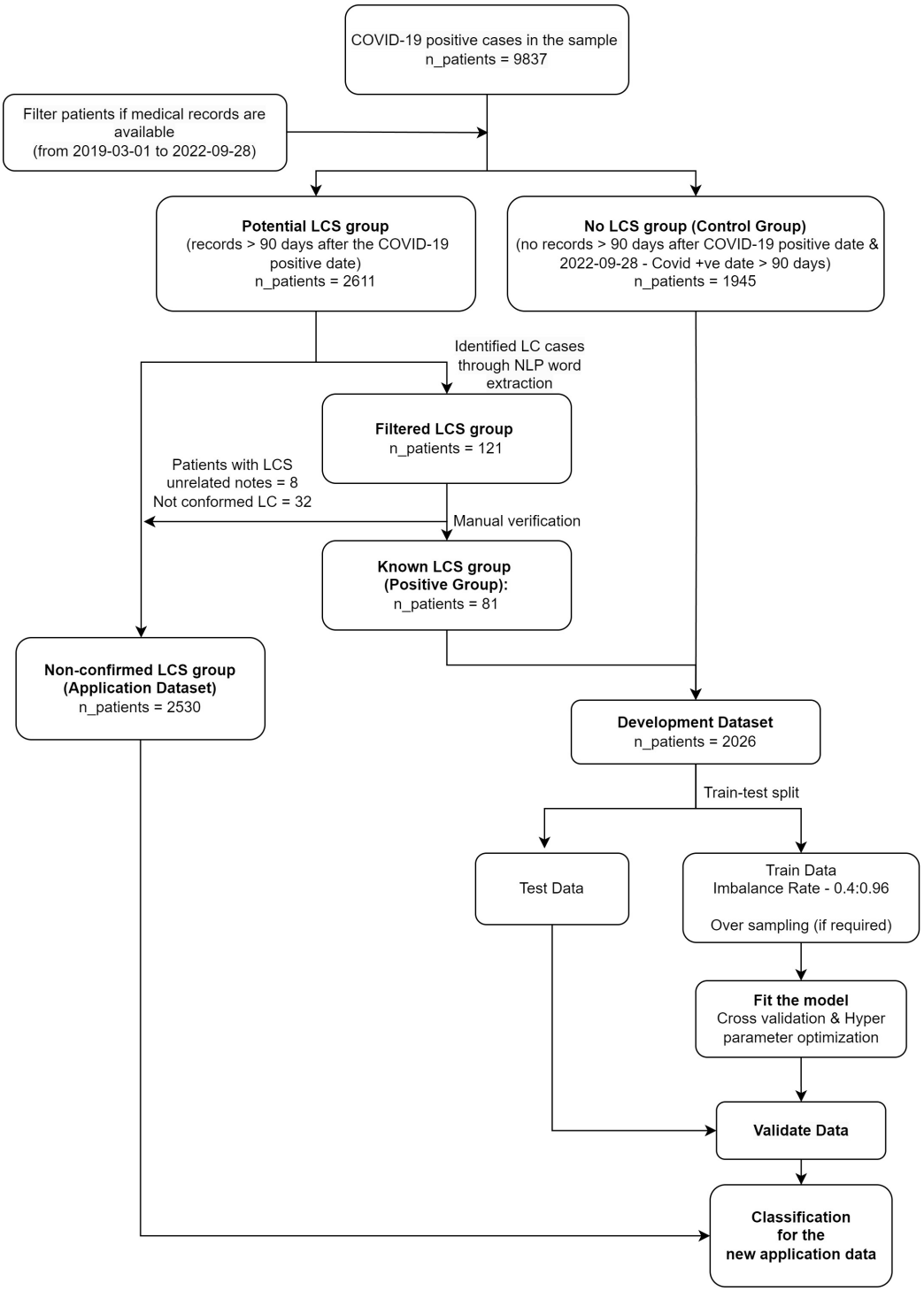


Figure 4.2: Process map

Overall, our approach to identifying a control group aimed to minimize confounding factors, ensuring that any associations observed between patient attributes, symptoms,

and LCS are specific to LCS and not merely reflective of general COVID-19 disease severity. Additionally, we identified 2611 patients with medical records extending beyond 90 days from the onset of COVID as potential LCS patients. Among this group, our method effectively identified 222 encounter notes belonging to 121 unique patients mentioning LCS by leveraging the linguistic patterns and medical domain knowledge (Fig. 4.2).

4.2.5 Symptom Extraction and Negation Identification

To improve our identification of symptoms related to LCS in medical data, we harnessed the power of natural language processing (NLP) and specifically focused on negation detection. We employed Python-based tools to streamline these processes. We determined whether the words mentioned were affirmed or negated with *Negex* (Negation Extraction) (Chapman et al.; 2001) and excluded those with negated LCS symptoms. The *Negex* algorithm, a widely used algorithm for negation detection in clinical texts, identifies phrases such as "no", "not," or "but" and subsequently identifies any medical terms that follow as negated (Chapman et al.; 2001). We also provided a separate list of terms that indicate negation (e.g., "abstain from," "other than," "excluding"). *Negex* examines the presence of these terms and carefully assesses the context to decide whether a given concept should be marked as negated. Using *Negex* allowed us to extract all negated medical terms from the electronic medical records (EMRs) of patients diagnosed with COVID-19. Concurrently, we extracted non-negated LCS-related symptoms by referring to a predetermined list of LCS symptoms as listed in (Matharaarachchi et al.; 2022).

However, for words such as taste, smell, appetite, and sleep, we are interested in the negation as loss of taste, smell, appetite, and lack of sleep were the symptoms of interest. In such cases, by considering negations, we could more precisely identify the presence or absence of symptoms in each patient's EMR.

4.2.6 Machine Learning Approach

To predict the LCS status of patients given several patient attributes, we used supervised machine learning. In supervised machine learning, a Development Dataset is used to train and test the predictive model. The Development Dataset set consists of labelled examples where the LCS status of patients is known. The patient attributes, such as age, gender, socioeconomic information, and pre- and post-COVID symptom profiles, are used as input features for the model. We split the Development Dataset into training and test sets, where the model learns patterns and relationships from the training data, allowing it to make predictions on new, unseen test data.

During training, the model adjusts its internal parameters to minimize the difference between the predicted LCS status and the actual labelled data. This process is often referred to as optimization or model fitting. Once trained, the model is applied to unseen the test data to validate model performance and to new patient data (Application dataset) to predict their LCS status based on their attributes.

It is important to note that the performance of the predictive model depends on the quality and representativeness of the training data, as well as the choice of the machine learning algorithm and its associated parameters. This study utilized three binary classification methods: Logistic Regression, Logistic Regression with Elastic Net Regularization for Classification, and Random Forest Classification. These methods were implemented using *glmnet* and *caret* R packages. Also, we used cross-validation and hyper-parameter optimization techniques to find the optimal set of parameters for each method.

4.2.6.1 Class Imbalance

Class imbalance is critical in predictive modelling and machine learning, particularly in scenarios where one class substantially outweighs the other in sample size. Class imbalance can have profound implications on model performance, as it tends to bias the learning process towards the majority class. In our study, the issue of class imbalance was of particular relevance as we sought to accurately identify individuals at risk of LCS. The LCS condition represents a minority within the broader context of COVID-19 cases.

We implemented two re-sampling techniques, Random Over-Sampling and Random Under-Sampling, to mitigate the potential impact of class imbalance, if present. In many supervised learning models, the frequency with which elements appear in the dataset can change the predictions of the model. By reinforcing those instances in the minority class, the model is able to learn better the features of minority class elements, which, in this context, represent LCS patients. With Random Over-Sampling, this was achieved by randomly duplicating samples from the LCS group, ensuring a more equitable representation of this critical class within our training dataset. Random Under-Sampling was designed to alleviate class imbalance by reducing the number of instances in the majority class (no-LCS cases). This involved randomly removing samples from the majority class, creating a more balanced distribution of classes.

The modification of our training set, influenced by both increased instances from Random Over-Sampling and reduced instances via Random Under-Sampling, significantly impacted the learning process of our machine learning models. Augmenting the minority class instances improved sensitivity, which is vital for precise medical diagnoses like LCS,

but introduced the potential risk of overfitting. Conversely, Random Under-Sampling, by reducing majority class instances, enabled the models to focus more on the minority class but posed the risk of information loss and reduced performance when encountering the majority class during testing. In response to these challenges, we produced a balanced class representation, aiming for a 50%:50% distribution within our dataset. Additionally, we selected machine learning techniques adept at mitigating overfitting concerns, ensuring that our models provided accurate predictions for LCS risk while maintaining the robustness and generalization capabilities when applied to unseen data.

4.2.6.2 Logistic Regression

Logistic regression (Weisberg; 2005) is a widely used statistical modelling method with a central role in predictive analytics and classification tasks. Its primary function is to model the probability of a binary outcome variable based on one or more predictor variables. The logistic function inherent to this model (Eq. 4.1) ensures that predicted probabilities are constrained within the range of 0 to 1.

Consider a training data set with n samples and m explanatory variables, $x_1, \ldots, x_m$. Since the logistic function is limited to taking values between 0 and 1. The logistic regression can be made by,

$$p(y = 1|x) = \frac{1}{1 + e^{-(b+w^T x)}} \tag{4.1}$$

where b is the model intercept, w represents coefficients $w_1, \ldots, w_m$ and $w^T x = w_1 x_1 + \ldots + w_m x_m$.

The logistic regression model estimates coefficients from the observed data through maximum likelihood estimation, enabling it to predict the probability of the binary outcome. This proficiency makes it well-suited for classifying new data points into one of the two specified categories. The model's versatility positions it as a robust tool for addressing classification tasks with several advantages, including simplicity, interpretability, and the ability to provide probabilistic outcomes. It finds widespread application across diverse domains, including healthcare, finance, and machine learning.

4.2.6.3 Logistic Regression with Elastic Net Regularization for Classification

Elastic Net classification is a statistical method used for classification problems, which combines the advantages of Lasso (Tibshirani; 1996) and Ridge (Hoerl and Kennard;

1970) regression methods. It was first introduced by Zou and Hastie (2005) as a regularization method for linear regression and later extended for classification problems. This method is based on a linear combination of L1 and L2 regularization penalties, which are applied to the coefficients of the linear classification model. The L1 penalty encourages sparsity in the coefficients, leading to feature selection and improved interpretability, while the L2 penalty helps to reduce overfitting by shrinking the coefficients towards zero. The elastic net regularization further enhances the model's capability to handle high-dimensional datasets and reduce the risk of overfitting all while maintaining its capacity to offer interpretable results and probabilistic outcomes.

The Elastic Net method can be solved using various optimization algorithms, such as coordinate descent, proximal gradient descent, and stochastic gradient descent. The algorithm's choice depends on the dataset's size and the solution's sparsity. Here, for this analysis, we used the "proximal Newton" algorithm for the optimization criterion for fast convergence. This makes repeated use of a quadratic approximation to the log-likelihood and then weighted coordinate descent on the resulting penalized weighted least-squares problem. These constitute an outer and inner loop known as iteratively reweighted penalized least squares.

4.2.6.4 Random Forest Classification

Random Forest classification (RFC) (Breiman; 2001) is a popular and effective machine learning algorithm widely used in statistical modelling for classification tasks. It belongs to the ensemble learning family and combines multiple decision trees to make accurate predictions. The underlying principle of RFC involves creating an ensemble of decision trees, each trained on a random subset of the training data and features. The RFC aggregates the predictions from individual trees to arrive at a final classification during prediction.

One of the critical advantages of RFC classification is its robustness to overfitting and noise and improved generalization performance. By aggregating predictions from multiple trees, RF reduces the impact of individual noisy or biased trees, leading to more robust and reliable predictions. Additionally, RFC can handle high-dimensional data, making it suitable for applications with many features. It automatically selects features by considering the most informative variables, thereby mitigating the curse of dimensionality. RFC also provides valuable insights into feature importance, aiding in variable selection and interpretability of the model.

4.2.6.5 Accuracy Measures

Sensitivity and specificity are crucial measures when predicting a disease, as they provide valuable insights into the accuracy and reliability of the prediction model. We assessed the overall performance of LCS prediction models by striking a balance between sensitivity and specificity.

Sensitivity measures the proportion of actual positive instances correctly identified by the model.

$$\text{Sensitivity } = \frac{TP}{(TP + FN)}$$

In the context of disease prediction, sensitivity indicates the model's ability to detect individuals who have the disease correctly. A high sensitivity value implies that the model has a low rate of false negatives, ensuring that true positive cases are not missed.

Specificity, on the other hand, quantifies the model's ability to identify negative instances correctly. It is the proportion of true negatives (TN) correctly classified among the sum of true negatives and false positives (FP).

$$\text{Specificity } = \frac{TN}{(TN + FP)}$$

In disease prediction, specificity reflects the model's ability to accurately classify individuals who do not have the disease as negative. A high specificity value indicates a low rate of false positives, which helps avoid unnecessary medical interventions, treatments, or additional testing for disease-free individuals. Specificity is particularly important in reducing unnecessary costs, patient anxiety, and potential adverse effects of false positive predictions.

The Area under the curve (AUC) of the Receiver operating characteristic curve (ROC) provides a way to assess the performance of a classification model by analyzing the trade-off between its true positive rate (TPR) and false positive rate (FPR). The ROC curve is created by plotting the TPR against the FPR at various classification thresholds. Each point on the ROC curve represents a different threshold setting of the model, indicating the performance at that threshold. The curve showcases the model's ability to discriminate between positive and negative instances across a range of threshold values. The AUC is a widely used measure to assess the discriminatory power of a classification model. It represents the probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative instance.

4.3 Statistical Analysis and Results

In our study, we evaluate three distinct machine learning algorithms alongside two alternative re-sampling techniques commonly used in the existing literature to predict individuals at risk of Long COVID Syndrome (LCS). Additionally, we consider two scenarios for including pre-COVID symptoms, one involving pre-COVID symptoms within 90 days of the COVID index date and another encompassing one year before the COVID index date. First, we divided our data set into training and test sets (80%:20%) to develop and evaluate our classification model for identifying LCS patients using patient attributes and symptoms. To perform the training-test split while preserving the class distribution in the response variable, we utilized the *createDataPartition* function from the *caret* package in R. Under/over-sampling was applied only to the training set, after the data split.

We then used 5-fold cross-validation to further evaluate the model's performance and mitigate over-fitting. The imbalance rate of the original data set was 4%:96% for the known LCS group and the no LCS control group (Fig. 4.2). We trained our model on the training set using classification algorithms and evaluated their performance on the test set. We used the *ROCR* package in R (Sing et al.; 2005) to establish a predictive probability threshold that maximized sensitivity and specificity. The best model was selected based on specificity, sensitivity, and AUC.

After analyzing the performance of the nine selected models based on sensitivity and specificity, we identified the best model by prioritizing sensitivity as the primary criterion. Accurately detecting at-risk LCS patients is of utmost importance, and maximizing sensitivity ensures a higher probability of correctly identifying individuals susceptible to LCS. By selecting the model that optimizes sensitivity, we aim to minimize false negatives and enhance the model's ability to capture as many true positive cases as possible.

Table 4.2 illustrates that the random over-sampled logistic regression model with elastic net regularization demonstrated the highest sensitivity among the selected models, making it the preferred choice for identifying at-risk LCS patients. Table 4.2 displays relevant counts and percentages, indicating how each method classified patients in the Application Dataset as LCS cases. It shows the total counts and percentages of the COVID-19-positive people projected to develop LCS. The selected method predicted 1043 (41%) of the patients from the application dataset as LCS. Along with the 81 (4%) initially known LCS patients from the development dataset, this accounts for 24.7% (1124 patients out of 4556) of patients within the COVID-19-positive group projected to develop LCS. This best-fitted model chooses post-COVID symptoms such as Breathing/lung issues, fatigue, other pains, chest pain, brain fog, dizziness and cough and age

group 70-79 as the most important eight features predicting the at-risk LCS group.

Table 4.2: Identified LCS patient counts and percentages with model accuracy measures

Pre-COVID symptom scenario	Dataset	Re-sampling Technique	Classification Method	No LCS Counts (%)	LCS Counts (%)	Total LCS Counts (%) (Development + Application)	AUC	Sensitivity	Specificity
90 days	Development Dataset			1945 (96%)	81 (4%)				
	Application Dataset	Baseline (Without Re-sampling)	Logistic	1657 (65%)	873 (35%)	954 (20.9%)	0.87	0.85	0.82
			Elastic Net	1857 (73%)	673 (27%)	754 (16.5%)	0.93	0.85	0.91
			Random Forest	1656 (65%)	874 (35%)	955 (21%)	0.93	0.9	0.85
		Random Over-Sampling	Logistic	1912 (76%)	618 (24%)	699 (15.3%)	0.88	0.85	0.86
			Elastic Net	1689 (67%)	841 (33%)	922 (20.2%)	0.93	0.9	0.83
			Random Forest	1779 (70%)	751 (30%)	832 (18.3%)	0.9	0.85	0.84
		Random Under-Sampling	Logistic	1480 (58%)	1050 (42%)	1131 (24.8%)	0.66	0.7	0.71
			Elastic Net	**1487 (59%)**	**1043 (41%)**	**1124 (24.7%)**	**0.94**	**0.95**	**0.81**
			Random Forest	1659 (66%)	871 (34%)	952 (20.9%)	0.93	0.9	0.86
1 year	Development Dataset			1592 (95%)	81 (5%)				
	Application Dataset	Baseline (Without Re-sampling)	Logistic	1825 (72%)	705 (28%)	786 (18.7%)	0.69	0.69	0.88
			Elastic Net	1459 (58%)	1071 (42%)	1152 (27.4%)	0.86	0.85	0.84
			Random Forest	1225 (48%)	1305 (52%)	1386 (33%)	0.84	0.85	0.79
		Random Over-Sampling	Logistic	1626 (64%)	904 (36%)	985 (23.4%)	0.66	0.69	0.79
			Elastic Net	1753 (69%)	777 (31%)	858 (20.4%)	0.75	0.77	0.84
			Random Forest	1347 (53%)	1183 (47%)	1264 (30.1%)	0.87	0.85	0.79
		Random Under-Sampling	Logistic	1594 (63%)	936 (37%)	1017 (24.2%)	0.79	0.69	0.84
			Elastic Net	1621 (64%)	909 (36%)	990 (23.6%)	0.79	0.85	0.83
			Random Forest	1816 (72%)	714 (28%)	795 (18.9%)	0.89	0.85	0.9

Following the manual refinement phase after the NLP approach, there were initial reservations regarding labelling 32 patients as LCS patients. Nevertheless, post-modeling, it became evident that the Elastic Net regression model, with under-sampling, accurately classified 27 of them, representing 84%, as LCS patients.

Fig. 4.3 showcases a strong consensus among all nine models when using the 90-day pre-COVID window. There is notable consistency, with at least seven models jointly predicting 22.3 percent of patients within the COVID-19-positive group who are poised to transition into LCS.

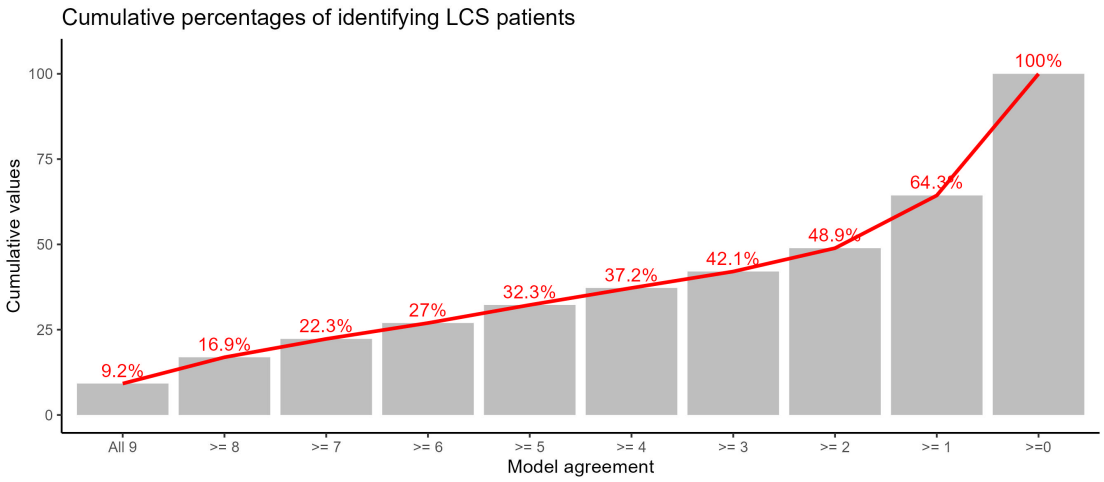


Figure 4.3: Agreement between nine models

Overall, LCS patients were found to have a higher likelihood of being female (59.7%) and aged between 50 and 59 years old (18.8%) (Table 4.3). These findings highlight the potential associations between demographic factors and the occurrence of LCS within the dataset.

Table 4.3: Demographic information of each patient group

Attribute	Development Dataset		Application Dataset		Rate Ratio		Overall	
	No LCS (Controls)	Known LCS	Predicted No LCS	Predicted LCS	No LCS	LCS	No LCS	LCS
	(N=1945)	(N=81)	(N=1487)	(N=1043)			(N=3432)	(N=1124)
Sex								
Female	1065 (54.8%)	45 (55.6%)	835 (56.2%)	626 (60.0%)	1.0	0.9	1900 (55.4%)	671 (59.7%)
Male	880 (45.2%)	36 (44.4%)	652 (43.8%)	417 (40.0%)	1.0	1.1	1532 (44.6%)	453 (40.3%)
SEFI								
Mean (SD)	-0.203 (0.680)	-0.191 (0.459)	-0.122 (0.724)	-0.101 (0.658)	1.7	1.9	-0.168 (0.701)	-0.107 (0.646)
Median [Min, Max]	-0.251 [-3.03, 3.16]	-0.178 [-1.98, 1.14]	-0.178 [-2.69, 3.88]	-0.178 [-3.40, 3.27]			-0.242 [-3.03, 3.88]	-0.178 [-3.40, 3.27]
Age Group								
<18	438 (22.5%)	s (s%)	236 (15.9%)	66 (6.3%)	1.4	-	674 (19.6%)	68 (6.0%)
18-29	367 (18.9%)	7 (8.6%)	301 (20.2%)	106 (10.2%)	0.9	0.8	668 (19.5%)	113 (10.1%)
30-39	323 (16.6%)	7 (8.6%)	225 (15.1%)	165 (15.8%)	1.1	0.5	548 (16.0%)	172 (15.3%)
40-49	259 (13.3%)	11 (13.6%)	198 (13.3%)	151 (14.5%)	1.0	0.9	457 (13.3%)	162 (14.4%)
50-59	253 (13.0%)	23 (28.4%)	215 (14.5%)	188 (18.0%)	0.9	1.6	468 (13.6%)	211 (18.8%)
60-69	167 (8.6%)	15 (18.5%)	160 (10.8%)	145 (13.9%)	0.8	1.3	327 (9.5%)	160 (14.2%)
70-79	74 (3.8%)	15 (18.5%)	79 (5.3%)	121 (11.6%)	0.7	1.6	153 (4.5%)	136 (12.1%)
80+	64 (3.3%)	s (s%)	73 (4.9%)	101 (9.7%)	0.7	-	137 (4.0%)	102 (9.1%)

[a] Rate ratio compares the prevalence among known vs predicting LCS patients (known/predicted), [b] 's' indicates the values below six which were suppressed.

The association between LCS and pre- and post-COVID symptoms in the final selected model is illustrated in Table 4.4 (pre-COVID symptoms) and Table 4.5 (post-COVID symptoms), respectively. These tables present the rate ratios for each symptom

in the LCS group compared to the reference group. Additionally, Table 4.6 (all COVID symptoms) provides the rate of change for each symptom in both the No LCS and LCS groups in the final model, spanning 90 days before the COVID index date.

Table 4.4: Pre-COVID symptom classification of each patient group

Symptom	Development Dataset		Application Dataset		Rate Ratio	
	No LCS Controls	Known LCS	Predicted No LCS	Predicted LCS	No LCS	LCS
	(N=1945)	(N=81)	(N=1487)	(N=1043)		
Ache	16 (0.8%)	0 (0%)	13 (0.9%)	16 (1.5%)	0.9	0.0
Allergy	21 (1.1%)	s (s%)	17 (1.1%)	21 (2.0%)	1.0	-
Appetite	s (s%)	0 (0%)	7 (0.5%)	s (s%)	-	-
Arm pain	24 (1.2%)	s (s%)	20 (1.3%)	41 (3.9%)	0.9	-
Back pain	12 (0.6%)	s (s%)	15 (1.0%)	39 (3.7%)	0.6	-
Body pain	s (s%)	0 (0%)	s (s%)	s (s%)	-	-
Breathing/lung issues	56 (2.9%)	7 (8.6%)	35 (2.4%)	99 (9.5%)	1.2	0.9
Cough	58 (3.0%)	s (s%)	45 (3.0%)	92 (8.8%)	1.0	-
Dysphagia	s (s%)	s (s%)	s (s%)	7 (0.7%)	-	-
Hand pain	0 (0%)	0 (0%)	0 (0%)	s (s%)	-	-
Headache	26 (1.3%)	s (s%)	35 (2.4%)	67 (6.4%)	0.5	-
Heart	23 (1.2%)	0 (0%)	24 (1.6%)	39 (3.7%)	0.8	0.0
Hemoptysis	s (s%)	0 (0%)	0 (0%)	s (s%)	-	-
Hot flushes	s (s%)	0 (0%)	0 (0%)	0 (0%)	-	-
Neck	14 (0.7%)	s (s%)	19 (1.3%)	34 (3.3%)	0.5	-
Pain	131 (6.7%)	15 (18.5%)	133 (8.9%)	207 (19.8%)	0.8	0.9
Phlegm	s (s%)	0 (0%)	s (s%)	13 (1.2%)	-	0.0
Shoulder pain	0 (0%)	0 (0%)	s (s%)	11 (1.1%)	-	0.0
Trauma	10 (0.5%)	s (s%)	10 (0.7%)	17 (1.6%)	0.7	-
Weight loss	8 (0.4%)	0 (0%)	8 (0.5%)	14 (1.3%)	0.8	0.0
Chest pain	41 (2.1%)	s (s%)	33 (2.2%)	67 (6.4%)	1.0	-
Fatigue	25 (1.3%)	s (s%)	22 (1.5%)	60 (5.8%)	0.9	-
Brain fog	7 (0.4%)	s (s%)	s (s%)	17 (1.6%)	-	-
Hoarse voice	s (s%)	0 (0%)	s (s%)	s (s%)	-	-
Ear pain	s (s%)	s (s%)	11 (0.7%)	8 (0.8%)	-	-
Joint pain	9 (0.5%)	s (s%)	13 (0.9%)	16 (1.5%)	0.6	-
Muscle pain	s (s%)	s (s%)	17 (1.1%)	31 (3.0%)	-	-
Mouth ulcer	s (s%)	0 (0%)	s (s%)	s (s%)	-	-
Numbness	16 (0.8%)	s (s%)	12 (0.8%)	20 (1.9%)	1.0	-
Leg pain	27 (1.4%)	s (s%)	25 (1.7%)	47 (4.5%)	0.8	-
Loss smell	s (s%)	0 (0%)	s (s%)	11 (1.1%)	-	0.0
Loss taste	s (s%)	0 (0%)	s (s%)	9 (0.9%)	-	0.0
Dizziness	15 (0.8%)	s (s%)	19 (1.3%)	47 (4.5%)	0.6	-

Table 4.5: Post-COVID symptom classification of each patient group

Symptom	Development Dataset		Application Dataset		Rate Ratio	
	No LCS Controls	Known LCS	Predicted No LCS	Predicted LCS	No LCS	LCS
	(N=1945)	(N=81)	(N=1487)	(N=1043)		
Ache	110 (5.7%)	16 (19.8%)	99 (6.7%)	188 (18.0%)	0.9	1.1
Allergy	132 (6.8%)	16 (19.8%)	142 (9.5%)	167 (16.0%)	0.7	1.2
Appetite	32 (1.6%)	s (s%)	32 (2.2%)	48 (4.6%)	0.7	-
Arm pain	166 (8.5%)	18 (22.2%)	207 (13.9%)	286 (27.4%)	0.6	0.8
Back pain	137 (7.0%)	12 (14.8%)	114 (7.7%)	199 (19.1%)	0.9	0.8
Body pain	13 (0.7%)	0 (0%)	7 (0.5%)	27 (2.6%)	1.4	0.0
Breathing/lung issues	325 (16.7%)	63 (77.8%)	0 (0%)	845 (81.0%)	-	1.0
Cough	346 (17.8%)	47 (58.0%)	221 (14.9%)	573 (54.9%)	1.2	1.1
Dysphagia	16 (0.8%)	s (s%)	16 (1.1%)	33 (3.2%)	0.7	-
Hand pain	s (s%)	0 (0%)	s (s%)	17 (1.6%)	-	0.0
Headache	185 (9.5%)	26 (32.1%)	168 (11.3%)	309 (29.6%)	0.8	1.1
Heart	163 (24%)	31 (38.3%)	146 (9.8%)	319 (30.6%)	0.9	1.3
Hemoptysis	s (s%)	s (s%)	s (s%)	25 (2.4%)	-	-
Hot flushes	s (s%)	0 (0%)	9 (0.6%)	8 (0.8%)	-	0.0
Neck	145 (7.5%)	25 (30.9%)	154 (10.4%)	227 (21.8%)	0.7	1.4
Pain	708 (36.4%)	64 (79.0%)	723 (48.6%)	811 (77.8%)	0.7	1.0
Phlegm	27 (1.4%)	11 (13.6%)	15 (1.0%)	68 (6.5%)	1.4	2.1
Shoulder pain	32 (1.6%)	s (s%)	36 (2.4%)	61 (5.8%)	0.7	-
Trauma	47 (2.4%)	8 (9.9%)	87 (5.9%)	116 (11.1%)	0.4	0.9
Weight loss	54 (2.8%)	8 (9.9%)	68 (4.6%)	116 (11.1%)	0.6	0.9
Chest pain	240 (12.3%)	42 (51.9%)	134 (9.0%)	520 (49.9%)	1.4	1.0
Fatigue	180 (9.3%)	51 (63.0%)	54 (3.6%)	473 (45.4%)	2.6	1.4
Brain fog	52 (2.7%)	19 (23.5%)	65 (4.4%)	123 (11.8%)	0.6	2.0
Hoarse voice	s (s%)	s (s%)	s (s%)	16 (1.5%)	-	-
Ear pain	63 (3.2%)	9 (11.1%)	65 (4.4%)	84 (8.1%)	0.7	1.4
Joint pain	79 (4.1%)	10 (12.3%)	87 (5.9%)	148 (14.2%)	0.7	0.9
Muscle pain	84 (4.3%)	12 (14.8%)	82 (5.5%)	174 (16.7%)	0.8	0.9
Mouth ulcer	s (s%)	s (s%)	9 (0.6%)	14 (1.3%)	-	-
Numbness	70 (3.6%)	10 (12.3%)	84 (5.6%)	135 (12.9%)	0.6	1.0
Leg pain	195 (10.0%)	26 (32.1%)	164 (11.0%)	283 (27.1%)	0.9	1.2
Loss smell	22 (1.1%)	s (s%)	20 (1.3%)	33 (3.2%)	0.8	-
Loss taste	16 (0.8%)	s (s%)	19 (1.3%)	28 (2.7%)	0.6	-
Dizziness	128 (6.6%)	26 (32.1%)	120 (8.1%)	296 (28.4%)	0.8	1.1

We also consider the observed phenomenon of higher percentage differences in symptom occurrence among LCS patients (Table 4.6). The analysis encompassed broad spectrum symptoms, revealing notable variations between the LCS and No LCS groups. Breathing/lung issues, other pains, chest pain, fatigue, and cough consistently displayed increased percentage differences in the LCS group. These findings align with the most

important features selected by our best-fitting model. The heightened percentage differences imply a more substantial shift in symptom occurrence post-COVID infection in this group, emphasizing that LCS is distinct from previous health conditions.

Table 4.6: Overall classification of symptoms with percentage difference

Symptom	No LCS			LC		
	Pre-COVID	Post-COVID	Percentage Difference	Pre-COVID	Post-COVID	Percentage Difference
Ache	29 (0.8%)	209 (6.1%)	5.3%	16 (1.4%)	204 (18.1%)	16.7%
Allergy	38 (1.1%)	274 (8.0%)	6.9%	22 (2.0%)	183 (16.3%)	14.3%
Appetite	10 (0.3%)	64 (1.9%)	1.6%	s (s%)	52 (4.6%)	-
Arm pain	44 (1.3%)	373 (10.9%)	9.6%	45 (4.0%)	304 (27.0%)	23%
Back pain	27 (0.8%)	251 (7.3%)	6.5%	43 (3.8%)	211 (18.8%)	15%
Body pain	s (s%)	20 (0.6%)	-	s (s%)	27 (2.4%)	-
Breathing/lung issues	91 (2.7%)	325 (9.5%)	6.8%	106 (9.4%)	908 (80.8%)	71.4%
Cough	103 (3.0%)	567 (16.5%)	13.5%	97 (8.6%)	620 (55.2%)	46.6%
Dysphagia	6 (s%)	32 (0.9%)	-	8 (0.7%)	37 (3.3%)	2.6%
Hand pain	0 (0%)	7 (0.2%)	0.2%	s (s%)	17 (1.5%)	-
Headache	61 (1.8%)	353 (10.3%)	8.5%	72 (6.4%)	335 (29.8%)	23.4%
Heart	47 (1.4%)	309 (9.0%)	7.6%	39 (3.5%)	350 (31.1%)	27.6%
Hemoptysis	s (s%)	9 (0.3%)	-	s (s%)	26 (2.3%)	-
Hot flushes	s (s%)	14 (0.4%)	-	0 (0%)	8 (0.7%)	0.7%
Neck	33 (1.0%)	299 (8.7%)	7.7%	37 (3.3%)	252 (22.4%)	19.1%
Pain	264 (7.7%)	1431 (41.7%)	34%	222 (19.8%)	875 (77.8%)	58%
Phlegm	9 (0.3%)	42 (1.2%)	0.9%	13 (1.2%)	79 (7.0%)	5.8%
Shoulder pain	s (s%)	68 (2.0%)	-	11 (1.0%)	67 (6.0%)	5%
Trauma	20 (0.6%)	134 (3.9%)	3.3%	18 (1.6%)	124 (11.0%)	9.4%
Weight loss	16 (0.5%)	122 (3.6%)	3.1%	14 (1.2%)	124 (11.0%)	9.8%
Chest pain	74 (2.2%)	374 (10.9%)	8.7%	69 (6.1%)	562 (50.0%)	43.9%
Fatigue	47 (1.4%)	234 (6.8%)	5.4%	61 (5.4%)	524 (46.6%)	41.2%
Brain fog	13 (0.4%)	117 (3.4%)	3%	18 (1.6%)	142 (12.6%)	11%
Hoarse voice	s (s%)	11 (0.3%)	-	s (s%)	17 (1.5%)	-
Ear pain	15 (0.4%)	128 (3.7%)	3.3%	9 (0.8%)	93 (8.3%)	7.5%
Joint pain	22 (0.6%)	166 (4.8%)	4.2%	17 (1.5%)	158 (14.1%)	12.6%
Muscle pain	21 (0.6%)	166 (4.8%)	4.2%	32 (2.8%)	186 (16.5%)	13.7%
Mouth ulcer	s (s%)	10 (0.3%)	-	s (s%)	15 (1.3%)	-
Numbness	28 (0.8%)	154 (4.5%)	3.7%	21 (1.9%)	145 (12.9%)	11%
Leg pain	52 (1.5%)	359 (10.5%)	9%	49 (4.4%)	309 (27.5%)	23.1%
Loss smell	8 (0.2%)	42 (1.2%)	1%	11 (1.0%)	35 (3.1%)	2.1%
Loss taste	s (s%)	35 (1.0%)	-	9 (0.8%)	31 (2.8%)	2%
Dizziness	34 (1.0%)	248 (7.2%)	6.2%	48 (4.3%)	322 (28.6%)	24.3%

4.4 Discussion

Our main goal was to develop a computational phenotype model that accurately identifies LCS cases within a substantial dataset of reported COVID-19 cases in Manitoba, Canada. We harnessed three distinct machine-learning techniques to achieve this: the logistic regression model, logistic regression with elastic net regularization, and the random forest classifier. For the model's training data, we focused on individuals within the cohort who were confirmed as LCS patients, a process facilitated through a natural language processing (NLP) word extraction method. The variables provided with the training set encompassed a broad spectrum of information, including socioeconomic information (SEFI), pre-existing symptoms, and COVID-19-related factors, such as all unique symptoms and conditions documented within the 90 days following the COVID-19 index date. By integrating this diverse dataset, we aimed to establish a robust and precise model for identifying individuals experiencing LCS in Manitoba.

This study extensively evaluated machine learning models to gauge their predictive capabilities and identify the most efficient approach for detecting LCS. Additionally, we employed two data re-balancing techniques, namely Random Over-Sampling and Random Under-Sampling, to tackle potential issues related to class imbalance. In particular, when applied to random under-sampled data, the logistic regression model with elastic net regularization showed superior performance to other methods, a topic we would like to explore further. This approach successfully identified 1124 (24.7%) LCS cases from the set of 4556 COVID-19 cases.

The logistic regression model with elastic net regularization brings several advantages in the context of LCS identification. Firstly, it incorporates feature selection by applying L1 and L2 regularization and identifying relevant predictors. This capability enhances the identification of the most informative variables, ultimately improving the model's interpretability. Secondly, elastic net regularization adeptly manages multicollinearity by combining L1 (Lasso) and L2 (Ridge) penalties. This approach effectively addresses issues from highly correlated predictor variables, supporting the model's stability. Lastly, the logistic regression model with elastic net regularization strikes a delicate balance between model complexity and overfitting, granting control over regularization parameters. This attribute encourages sparsity within the model, enhancing its ability to generalize to unseen data and, in turn, elevating the accuracy of LCS case identification. Collectively, these attributes establish the logistic regression model with elastic net regularization as an apt and practical approach for LCS detection in this study, affirming the findings reported elsewhere (Binka et al.; 2022).

In addition, the Random Under-Sampling technique merits attention for mitigating

class imbalance. Randomly removing instances from the majority class reduces computational complexity and helps alleviate the risk of overfitting. This technique proves particularly valuable when working with a reduced dataset, as it maintains the model's resilience while addressing class imbalance concerns.

Concerns have arisen regarding the representativeness of cases in community-based studies, raising questions about their applicability to the broader population (Sorensen et al.; 1998). To address this concern, a population-based study encompassing all encounter notes within Manitoba would yield more accurate and inclusive inferences. It is important to note that the clinical notes in this study were drawn from physicians who had consented to the extraction of their EMRs for research purposes, introducing the potential for sample bias and limiting the sample size. Also, an essential consideration is the need for appropriate control groups to accurately identify and measure symptoms attributable to LCS, particularly when assessing its long-term consequences.

As we continue to identify additional data sources, our models can undergo retraining and fine-tuning to cater to the specific requirements of individual studies.

4.5 Conclusion

Utilizing natural language processing to identify confirmed LCS patients initially and applying a logistic regression model with elastic net regularization combined with random under-sampling to classify individuals at risk of LCS addresses a significant challenge within the healthcare sector. The outcomes of this approach underscore its potential to accurately identify individuals prone to LCS, as indicated by the noteworthy accuracy metrics: sensitivity of 0.95, specificity of 0.81, and AUC of 0.94. This cohort of LCS patients, established through this methodology, can serve as a foundational resource for conducting robust assessments of the clinical course of LCS and offers vital concrete data for informed decision-making in healthcare.

Bibliography

Amin-Chowdhury, Z. and Ladhani, S. N. (2021). Causation or confounding: why controls are critical for characterizing long covid, *Nature medicine* **27**(7): 1129–1130.

A systematic investigation of Manitoba's Provincial Laboratory data (2012). Manitoba Centre for Health Policy, Dept. of Community Health Sciences, Faculty of Medicine, University of Manitoba, Winnipeg.

Binka, M., Klaver, B., Cua, G., Wong, A. W., Fibke, C., Velásquez García, H. A., Adu, P., Levin, A., Mishra, S., Sander, B., Sbihi, H. and Janjua, N. Z. (2022). An elastic net regression model for identifying long covid patients using health administrative data: A population-based study, *Open forum infectious diseases* **9**(12): ofac640–ofac640.

Birtwhistle, R. V. (2011). Canadian primary care sentinel surveillance network: a developing resource for family medicine and public health, *Canadian family physician* **57**(10): 1219–1220.

Blomberg, B., Mohn, K. G.-I., Brokstad, K. A., Zhou, F., Linchausen, D. W., Hansen, B.-A. and Lartey, S. (2021). Long covid in a prospective cohort of home-isolated patients, *Nature medicine* **27**(9): 1607–1613.

Breiman, L. (2001). Random forests, *Machine learning* **45**(1): 5–32.

Bull-Otterson, L., Baca, S., Saydah, S., Boehmer, T. K., Adjei, S., Gray, S. and Harris, A. M. (2022). Post–covid conditions among adult covid-19 survivors aged 18–64 and $\geq$ 65 years — united states, march 2020–november 2021.

Carfì, A., Bernabei, R. and Landi, F. (2020). Persistent symptoms in patients after acute covid-19, *JAMA : the journal of the American Medical Association* **324**(6): 603–605.

Celi, L. A., Majumder, M. S., Ordóñez, P., Osorio, J. S., Paik, K. E. and Somai, M. (2020). Introduction to clinical natural language processing with python, *Leveraging Data Science for Global Health*, Springer International Publishing AG, Switzerland.

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F. and Buchanan, B. G. (2001). A simple algorithm for identifying negated findings and diseases in discharge summaries, *Journal of biomedical informatics* **34**(5): 301–310.

Characterizing Long COVID in an international cohort: 7 months of symptoms and their impact (2021). *EClinicalMedicine* **38**: 101019–101019.

Chateau, D., Metge, C., Prior, H. and Soodeen, R.-A. (2012). Learning from the census: The socio-economic factor index (sefi) and health outcomes in manitoba, *Canadian journal of public health* **103**(8 Suppl 2): S23–S27.

Crook, H., Raza, S., Nowell, J., Young, M. and Edison, P. (2021). Long covid—mechanisms, risk factors, and management, *BMJ (Online)* **374**: n1648–n1648.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **12**(1): 55–67.

Huang, C., Huang, L., Wang, Y., Li, X., Ren, L., Gu, X., Kang, L., Guo, L., Liu, M., Zhou, X., Luo, J., Huang, Z., Tu, S., Zhao, Y., Chen, L., Xu, D., Li, Y., Li, C., Peng, L., Li, Y., Xie, W., Cui, D., Shang, L., Fan, G., Xu, J., Wang, G., Wang, Y., Zhong, J., Wang, C., Wang, J., Zhang, D. and Cao, B. (2021). 6-month consequences of covid-19 in patients discharged from hospital: a cohort study, *The Lancet (British edition)* **397**(10270): 220–232.

Keselj, Vlado (Review of: Jurafsky, D. and Martin, J. H. (2009). Speech and language processing (second edition), *Computational Linguistics* **35**(3): 463–466.

Margalit, I., Yelin, D., Sagi, M., Rahat, M. M., Sheena, L., Mizrahi, N., Gordin, Y., Agmon, H., Epstein, N. K., Atamna, A., Tishler, O., Daitch, V., Babich, T., Abecasis, D., Yarom, Y., Kazum, S., Shitenberg, D., Baltaxe, E., Elkana, O., Shapira-Lichter, I., Leibovici, L. and Yahav, D. (2022). Risk factors and multidimensional assessment of long coronavirus disease fatigue: A nested case-control study, *Clinical infectious diseases* **75**(10): 1688–1697.

Matharaarachchi, S., Domaratzki, M., Katz, A. and Muthukumarana, S. (2022). Discovering long covid symptom patterns: Association rule mining and sentiment analysis in social media tweets, *JMIR formative research* **6**(9): e37984–e37984.

Moldofsky, H. and Patcai, J. (2011). Chronic widespread musculoskeletal pain, fatigue, depression and disordered sleep in chronic post-sars syndrome; a case-controlled study, *BMC neurology* **11**(1): 37–37.

Nalbandian, A., Sehgal, K., Gupta, A., Madhavan, M. V., McGroder, C., Stevens, J. S., Cook, J. R., Nordvig, A. S., Shalev, D., Sehrawat, T. S., Ahluwalia, N., Bikdeli, B., Dietz, D., Der-Nigoghossian, C., Liyanage-Don, N., Rosner, G. F., Bernstein, E. J., Mohan, S., Beckley, A. A., Seres, D. S., Choueiri, T. K., Uriel, N., Ausiello, J. C., Accili, D., Freedberg, D. E., Baldwin, M., Schwartz, A., Brodie, D., Garcia, C. K., Elkind, M. S. V., Connors, J. M., Bilezikian, J. P., Landry, D. W. and Wan, E. Y. (2021). Post-acute covid-19 syndrome, *Nature medicine* **27**(4): 601–615.

Pfaff, E. R., Girvin, A. T., Bennett, T. D., Bhatia, A., Brooks, I. M., Deer, R. R., Dekermanjian, J. P., Jolley, S. E., Kahn, M. G., Kostka, K., McMurry, J. A., Moffitt, R., Walden, A., Chute, C. G., Haendel, M. A., Bramante, C., Dorr, D., Morris, M., Parker, A. M., Sidky, H., Gersing, K., Hong, S. and Niehaus, E. (2022). Identifying who has long covid in the USA: a machine learning approach using n3c data, *The Lancet (British edition)* **4**(7): e532–e541.

Sing, T., Sander, O., Beerenwinkel, N. and Lengauer, T. (2005). Rocr: visualizing classifier performance in r, *Bioinformatics* **21**(20): 7881.

Sorensen, G., Emmons, K., Hunt, M. K. and Johnston, D. (1998). Implications of the results of community intervention trials, *Annual review of public health* **19**(1): 379–416.

Sudre, C. H., Murray, B., Varsavsky, T., Graham, M. S., Penfold, R. S., Bowyer, R. C., Pujol, J. C., Klaser, K., Antonelli, M., Canas, L. S., Molteni, E., Modat, M., Jorge Cardoso, M., May, A., Ganesh, S., Davies, R., Nguyen, L. H., Drew, D. A., Astley, C. M., Joshi, A. D., Merino, J., Tsereteli, N., Fall, T., Gomez, M. F., Duncan, E. L., Menni, C., Williams, F. M. K., Franks, P. W., Chan, A. T., Wolf, J., Ourselin, S., Spector, T. and Steves, C. J. (2021). Attributes and predictors of long covid, *Nature Medicine* **27**(4): 626–631.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B, Methodological* **58**(1): 267–288.

Weisberg, S. (2005). *Applied Linear Regression*, John Wiley and Sons, Incorporated, Hoboken.

Yelin, D., Margalit, I., Nehme, M., Bordas-Martínez, J., Pistelli, F., Yahav, D., Guessous, I., Durà-Miralles, X., Carrozzi, L., Shapira-Lichter, I., Vetter, P., Peleato-Catalan, D., Tiseo, G., Wirtheim, E., Kaiser, L., Gudiol, C., Falcone, M., Leibovici, L. and Group, O. B. O. T. L. R. (2022). Patterns of long covid symptoms: A multi-center cross sectional study, *Journal of clinical medicine* **11**(4): 898–.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society. Series B, Statistical methodology* **67**(2): 301–320.

Chapter 5

Enhancing SMOTE for Imbalanced Data with Abnormal Minority Instances

The content of this chapter is a second revision of the manuscript text for publication under the following citation:

Matharaarachchi, S., Domaratzki, M. and Muthukumarana, S. (2024). Enhancing smote for imbalanced data with abnormal minority instances, Machine learning with applications 18: 100597.

Enhancing SMOTE for Imbalanced Data with Abnormal Minority Instances

Surani Matharaarachchi[a,*], Mike Domaratzki[b], Saman Muthukumarana[a]

[a]*Department of Statistics, University of Manitoba, Winnipeg, MB, Canada*
[b]*Department of Computer Science, Western University, London, ON, Canada*

Abstract

Imbalanced datasets are frequent in machine learning, where certain classes are markedly underrepresented compared to others. This imbalance often results in sub-optimal model performance, as classifiers tend to favour the majority class. A significant challenge arises when abnormal instances, such as outliers, exist within the minority class, diminishing the effectiveness of traditional re-sampling methods like the Synthetic Minority Oversampling Technique (SMOTE). This manuscript addresses this critical issue by introducing four novel SMOTE extensions: Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE, and BGMM SMOTE. These methods leverage a weighted average of neighbouring instances to enhance the quality of synthetic samples and mitigate the impact of outliers. Comprehensive experiments conducted on diverse simulated and real-world imbalanced datasets demonstrate that the proposed methods improve classification performance compared to the original SMOTE and its most competitive variants. Notably, we demonstrate that Dirichlet ExtSMOTE outperforms most other proposed and existing SMOTE variants in terms of achieving better F1 score, MCC, and PR-AUC. Our results underscore the effectiveness of these advanced SMOTE extensions in tackling class imbalance, particularly in the presence of abnormal instances, offering robust solutions for real-world applications.

Keywords: *Class Imbalance, Abnormal Instances, Inverse Distance, SMOTE, Bayesian Gaussian Mixture Models, Dirichlet Distribution*

---

*Corresponding Author

Email addresses: matharas@myumanitoba.ca (Surani Matharaarachchi), mdomarat@uwo.ca (Mike Domaratzki), saman.muthukumarana@umanitoba.ca (Saman Muthukumarana)

5.1 Introduction

The issue of class imbalance poses a common challenge in machine learning, where instances are unevenly distributed across different classes. This issue is particularly prominent in real-world applications such as medical diagnosis (Mazurowski et al.; 2008; Matharaarachchi et al.; 2021), fraud detection (Yang et al.; 2009), and churn prediction (Zhu et al.; 2018), where minority classes are of particular interest. Imbalanced data introduces substantial obstacles for machine learning algorithms, leading to the development of biased models that perform poorly on the minority class.

In these imbalanced datasets, one or more classes contain significantly higher or lower instances than others. The class with more instances is typically called the majority class, while the one with fewer instances is termed the minority class.

This type of data can be categorized based on its imbalance ratio (IR) (Fernández et al.; 2010), which is defined as the ratio of the number of examples in the majority class to those in the minority class. A higher imbalance ratio indicates a greater degree of imbalance between the classes. For example, if the class imbalance is $0.25 : 0.75$, then the IR is 3. In contrast, if the imbalance is $0.05 : 0.95$, the IR is 19.

This imbalance poses difficulties for traditional machine learning methods, making it challenging to learn from such datasets effectively. When abnormal instances are present in the minority class, the problem is exacerbated. Outliers and noisy data can distort the feature distribution, leading to the generation of unrepresentative synthetic samples. This, in turn, can result in models that are biased towards the majority class and unable to predict the minority class accurately.

Many solutions have been proposed to address class imbalance; however, a substantial research gap still exists in effectively handling abnormal instances within the minority class while employing re-balancing techniques. Most existing methods struggle to maintain prediction accuracy when generating synthetic data in the presence of these abnormal instances.

In this paper, we introduce four novel methods for handling dataset imbalance for datasets with abnormal instances: Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE, and BGMM SMOTE. We conducted experiments supported by simulated and

real-world data, with a particular focus on the presence of abnormal instances in the minority class, demonstrating the effectiveness of these proposed solutions in enhancing classification performance. Classification was performed using three classifiers: Logistic Regression (Weisberg; 2005), K-Nearest Neighbors (KNN) (Cover and Hart; 1967), and Random Forest (Breiman; 2001). We evaluated model performance using various accuracy measures, including F1-Score, PR-AUC, and MCC. The four proposed methods provide alternative approaches that may be competitive with each other under different circumstances.

The primary objectives of this manuscript are as follows:

- To provide insights into the challenges of abnormal instances within the minority class when using SMOTE and some popular existing extensions.
- To propose several innovative strategies capable of mitigating the adverse effects of abnormal instances on class imbalance data.

The structure of this manuscript is organized as follows: Section 2 delves deeper into the limitations of SMOTE in the presence of abnormal instances, offering a comprehensive understanding of the issue. It also explores various strategies and techniques proposed to address this challenge. Section 3 presents empirical results from experiments conducted to validate the effectiveness of these strategies. Section 4 compares the results of each method using real-world application datasets. Finally, Section 5 of this paper is evaluated with a discussion of its contributions and limitations, whereas Section 6 provides the conclusion.

5.2 Literature Review

To address the challenge of class imbalance, researchers have developed oversampling and undersampling techniques. These methods create samples for the minority class or remove samples from the majority class, helping to balance the dataset by equalizing the number of data points across all classes. One notable approach in this domain is the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al.; 2002), which aims to address class distribution imbalance by generating artificial samples for the minority class. SMOTE has proven successful in enhancing classifier performance on imbalanced datasets by augmenting the representation of the minority class and expanding the training dataset.

Nonetheless, SMOTE is not without its limitations. It frequently fails to account for crucial factors, including the distribution of minority classes and hidden noise within the

dataset, as noted by Hu et al. (2009). One notable drawback is SMOTE's inclination to overly generalize the minority class, resulting in misclassifications within the majority class and disturbing the overall equilibrium of the model, as emphasized by Blagus and Lusa (2013). While SMOTE has proven effective in various applications, its performance can also be compromised by abnormal instances. Abnormal instances can distort the synthetic sample generation process, leading to suboptimal results.

Researchers have proposed several extensions to overcome the limitations of the original SMOTE technique. Some of these methods focus on enhancing the generation of synthetic data by integrating SMOTE with other oversampling techniques. Notable examples include combining SMOTE with techniques like particle swarm optimization (Gao et al.; 2011), kernel-based approaches (Mathew et al.; 2015), and Boosting (Chawla et al.; 2003). Bej et al. (2021) addressed the issue of over-generalizing the minority class by SMOTE by employing Localized Random Affine Shadowsampling (LoRAS).

To address the limitations of SMOTE in the presence of outliers, researchers have developed various extensions and adaptations. These variants aim to create synthetic samples that are robust to the influence of outliers, focusing on more accurate and reliable models.

Borderline-SMOTE (Han et al.; 2005) is a variant that focuses on the borderline instances of the minority class. Generating synthetic samples near the boundary between classes helps mitigate the impact of outliers, providing better class separation.

ADASYN (Adaptive Synthetic Sampling) method (He et al.; 2008) adapts the number of synthetic samples generated for each minority instance based on the local distribution and classification difficulty. Synthetic observations are created based on the difficulty level of learning specific instances within the minority class.

SVM-SMOTE (Suh et al.; 2017) integrates SMOTE with the Support Vector Machine (SVM) algorithm to guide the synthetic sample generation. SVM identifies the support vectors that define the decision boundary between classes. Synthetic samples are generated along the support vectors, ensuring they are close to the decision boundary.

SMOTE-IPF (Sáez et al.; 2015) incorporates an iterative ensemble-based noise filtering mechanism called Iterative-Partitioning Filter to handle noisy and borderline examples. After each iteration of synthetic sample generation, the filter evaluates the dataset and removes outliers. This iterative process continues until no further noise is detected, resulting in a synthetic dataset that is free from outliers and more representative of the minority class distribution.

Outlier-SMOTE, proposed by Turlapati and Prusty (2020), adopts a strategy of oversampling each data point based on its distance from other data points.

SMOTE-LOF (SMOTE combined with Local Outlier Factor) (Asniar et al.; 2022) enhances the synthetic sample generation process by integrating LOF to identify and exclude outliers in the minority class. LOF calculates the local density deviation of data points to detect outliers. By removing these outliers before generating synthetic samples, SMOTE-LOF ensures that the synthetic data is more representative of the minority class's true distribution. This approach prevents the generation of synthetic samples around outliers, which could lead to poorly generalized synthetic samples and negatively impact the classifier's performance.

SMOTE-TomekLinks (Batista et al.; 2004) integrates SMOTE with Tomek Links, a data cleaning technique that removes overlapping instances between classes. After generating synthetic samples using SMOTE, Tomek Links are applied to identify and remove instances near the decision boundary, likely to be noise or outliers. This combined approach enhances the dataset by cleaning outliers and reducing class overlap, leading to better classifier performance.

SMOTE-ENN (Edited Nearest Neighbors) (Batista et al.; 2004) combines SMOTE with ENN, which removes instances misclassified by their nearest neighbors, effectively cleaning the dataset of noise and outliers. After generating synthetic samples with SMOTE, ENN filters out unreliable samples, ensuring that outliers in the minority class are not retained. This hybrid method improves the robustness of the synthetic dataset.

MWMOTE (Majority Weighted Minority Oversampling Technique) (Barua et al.; 2014) identifies hard-to-learn minority instances, often outliers, and assigns them higher weights during the oversampling process. However, it avoids generating synthetic samples directly around these outliers, focusing instead on more informative minority instances. This reduces the influence of outliers on the synthetic dataset, leading to better classification performance.

DBSMOTE (Density-Based SMOTE) (Bunkhumpornpat et al.; 2012) uses a density-based clustering approach to identify and handle outliers in the minority class. By applying a density-based clustering algorithm, DBSMOTE identifies sparse regions where outliers are likely to exist and generates synthetic samples within denser regions. This method ensures that outliers do not adversely affect the synthetic data generation process.

DSMOTE (Mahmoudi et al.; 2014) is similar to DBSMOTE in using density information to guide synthetic sample generation. It focuses on generating synthetic samples

in denser regions of the minority class, avoiding sparse areas where outliers are more likely to be found. This density-based approach helps create a more representative and outlier-free synthetic dataset.

ProWSyn (Proximity Weighted Synthetic Oversampling) (Barua et al.; 2013) uses proximity information to guide the oversampling process. It assigns weights to minority instances based on their proximity to other instances, with outliers receiving lower weights. This weighted approach minimizes the impact of outliers on the synthetic dataset, ensuring the synthetic data remains representative of the minority class.

Cluster SMOTE (Cieslak et al.; 2006) applies clustering algorithms to group minority instances before performing oversampling. Clusters containing outliers are either excluded or given less importance in the oversampling process. By generating synthetic samples within well-defined clusters and avoiding outlier-prone clusters, Cluster SMOTE ensures that the synthetic data accurately represents the minority class distribution.

Gaussian SMOTE (Lee and Kim; 2017) generates synthetic samples based on Gaussian distributions fitted to the minority class instances. Outliers, which do not conform to the main data distribution, have less influence on the Gaussian parameters, resulting in fewer synthetic samples being generated around them. This method effectively mitigates the impact of outliers on the synthetic dataset.

One key insight from analyzing existing SMOTE variants is that many methods clean the data by removing outliers from the original dataset. While this may seem beneficial, it necessitates the addition of more synthetic instances to rebalance the dataset. This approach can overlook the valuable information that outliers might provide, potentially reducing the robustness and accuracy of the model's predictions.

By exploring the outcomes of some of the most common and competitive extensions (Kovács; 2019) alongside our novel approaches, we aim to gain valuable insights into improving the handling of imbalanced datasets in machine learning.

5.3 Methodology

5.3.1 Abnormal Instances

In binary classification, abnormal instances refer to data points that deviate significantly from most of the dataset or are incorrectly labelled. These instances can be outliers, noise, or mislabeled data, causing challenges in building accurate predictive models (Fig. 5.1).

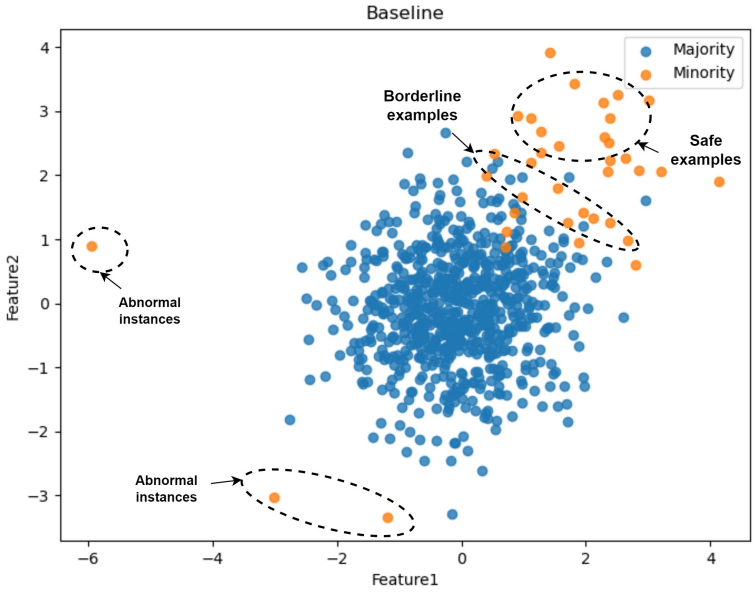


Figure 5.1: Abnormal instances in the minority class

An outlier is "an observation or data point that significantly deviates from the majority of other observations in a dataset, often arousing suspicion that a different or abnormal process may have generated it" (Hawkins; 1980). Noise refers to data points deviating from general patterns, while mislabeled data are incorrectly categorized instances.

In this paper, we define the outlier ratio (OR) as the ratio of the number of outlier instances to the number of inlier instances. A higher outlier ratio indicates a greater prevalence of outliers relative to the inliers within the dataset. For example, an OR of approximately 0.0526 (≈ 0.05) corresponds to a 5% prevalence of outliers, and an OR of approximately 0.1111 (≈ 0.1) corresponds to a 10% prevalence of outliers.

Abnormal instances in a dataset may reflect legitimate extreme observations due to random fluctuations, representing inherent sampling characteristics. These should be retained and treated equally in analysis. However, these abnormal instances might not be useful for extensively generating new synthetic instances. For instance, abnormal instances may distort the dataset in medical diagnosis, misleading algorithms and biasing predictions. These instances, residing at the data's edges, challenge SMOTE's efficacy by generating unrepresentative synthetic samples. Therefore, effective strategies are crucial to applying SMOTE and ensuring reliable predictions.

5.3.2 Overview of SMOTE

SMOTE (Chawla et al.; 2002) is a widely used and effective approach for tackling class imbalance in classification datasets. SMOTE addresses the issue of data scarcity in the minority class by generating synthetic samples that bridge the gap between minority class instances in feature space. The fundamental idea behind SMOTE is to create new synthetic instances by interpolating feature values between randomly selected minority class instances and one of their k-nearest neighbours.

The SMOTE algorithm (Alg. 2) follows a simple and intuitive process. For each minority class instance, it selects k-nearest neighbours from the minority class instances. It then creates synthetic instances along the line segments connecting the selected instance with one of its k-nearest neighbours. These synthetic instances introduce new data points in the feature space and increase the representation of the minority class.

Algorithm 2 Original SMOTE

Require: $X \in \mathbb{R}^{n \times p}$ the features.
Require: $Y \in \{0,1\}^n$ the binary class label outputs.
Require: $k \in \mathbb{N}$ the number of neighbors to select for the k-Nearest Neighbors.
Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{0,1\}^q$ with q points created.

1: Denote by S_1 the number of points labeled as the minority class and S_0 the number of points labeled as the majority class.
2: Initialize X_{new} and Y_{new} as empty vectors.
3: **while** $S_1 < S_0$ **do**
4: Filter $\mathcal{D} = \{X_i | Y_i = 1\}$, the set of points labeled as minority class 1.
5: Randomly choose $r \in \mathcal{D}$ and find the indices of its k nearest neighbors.
6: Randomly choose an index r_2 among these neighbors.
7: $x^{new} \leftarrow \alpha \times x_{r_1} + (1 - \alpha) \times x_{r_2}$ with $\alpha \in [0,1]$ randomly chosen.
8: $y^{new} \leftarrow 1$
9: $S_1 = S_1 + 1$
10: Append x^{new} to X_{new}, append y^{new} to Y_{new}
11: return X_{new}, Y_{new}

The main goal of SMOTE is to balance classes by creating new similar samples for the minority class. However, when abnormal instances are part of this process, SMOTE might prioritize replicating these abnormalities instead of the typical minority class examples. This can lead to noisy synthetic samples that don't represent the true characteristics of the minority class well. Abnormal instances could make SMOTE generate new samples in data areas that don't truly represent the minority class, affecting the model's ability to generalize accurately and harming its performance. Fig. 5.2 shows the SMOTE-generated data based on the initial dataset displayed in Fig. 5.1. Within

this visualization, there are three noticeable abnormal instances, and it's clear how these instances create an unusual data bridge between the minority data points and the abnormal ones.

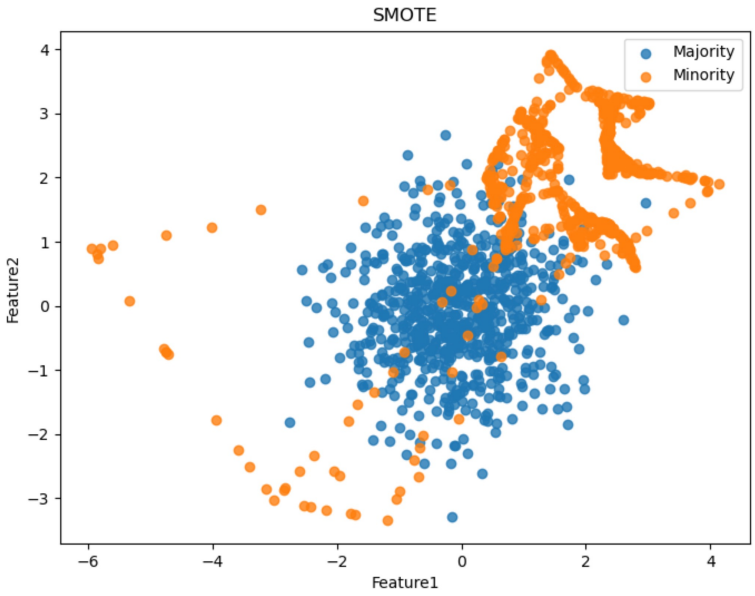
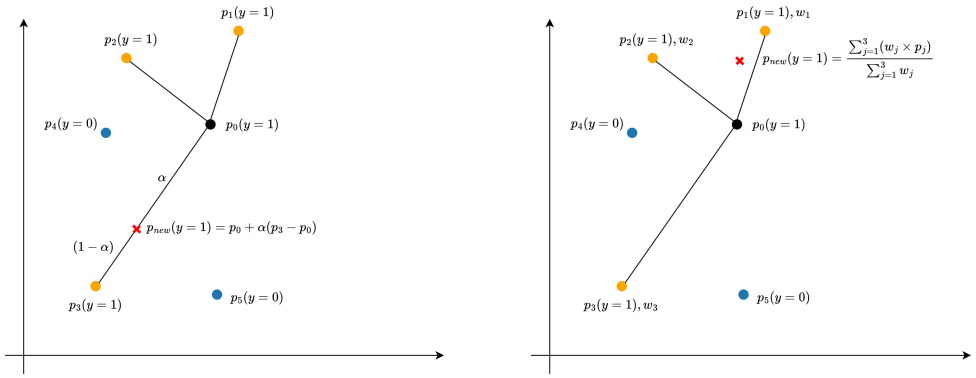


Figure 5.2: Re-sampled data with SMOTE

5.3.3 Proposed Methods - Based on Weighted Average

As a solution, we propose new over-sampling techniques called Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE and BGMM SMOTE, Weighted average-based Minority Over-sampling TEchniques, which use weighted averages of nearest neighbours instead of a basic linear combination, allowing for learning from a more extensive set of nearest neighbours. Fig. 5.3 illustrates how our method modifies the synthetic data generation process compared to the standard SMOTE approach. In the following sections, we introduce various methods for determining appropriate weights w_j, to improve the model's ability to handle abnormal instances more effectively.



(a) SMOTE data generation

(b) Proposed data generation

Figure 5.3: Data generation mechanisms

5.3.4 The Distance Measure

Distance is used to quantify the similarity between instances. Most of the extensions of SMOTE, which use distance metrics, consider the distance between the minority class instances to generate synthetic instances (Han et al.; 2005; He et al.; 2008; Mahmoudi et al.; 2014). According to Feng et al. (2022), the selection of this distance does not impact the overall performances of existing SMOTE-based techniques, nor does the distance metric.

In contrast to these extensions, our approach involves leveraging the distances between the median centroid and the chosen nearest neighbours to determine the weights. The fundamental idea is to provide higher weights to instances closer in feature space, thus giving them more influence in the synthetic instance creation. This strategy proves particularly valuable in the presence of abnormal instances within the minority class, where minimizing their influence is essential. We opt for the median centroid, known for its robustness in the face of outliers. Even if the selected point is not abnormal, it is important to note that the generation of the synthetic data point will be distributed fairly around its neighbouring data points.

We initiate the process by determining the median centroid of the minority class, represented as μ, which is essentially the point that serves the “center" of these instances in a multidimensional space. It is calculated as the median of each feature (dimension) separately. After that, we identify the nearest neighbours for a random instance. These nearest neighbours, denoted as x_i, manifest as feature vectors that capture the attribute values of these proximate data points. Each x_i vector represents the feature values of the i^{th} nearest neighbour. To calculate the distance between μ and each nearest neighbour

x_i, we use the Euclidean distance (L2 norm), a measure of the straight-line distance between two points in Euclidean space (Eq. 5.1).

$$d(\mu, x_i) = \sqrt{\sum_{j=1}^{n} (\mu_j - x_{ij})^2} \tag{5.1}$$

By taking the inverse of distances, instances closer to the median centroid receive higher weight, ensuring that the synthetic samples are more representative of the central tendency of the minority class and effectively minimize the effect of abnormal instances in the minority data. It enhances the robustness and accuracy of synthetic sample generation, contributing to improved classification performance in the presence of imbalanced datasets with abnormal instances.

5.3.5 Distance ExtSMOTE

In the first proposed method, Distance ExtSMOTE, the normalized inverse distance between the median centroid and the chosen k nearest neighbours is directly used as the weights (Alg. 3).

Fig. 5.4 demonstrates generating a new point using our suggested technique. In Fig. 5.4b, a closer view of the data generation in Fig. 5.4a is presented. In a typical SMOTE scenario, the new point usually falls midway along one of the three lines. Yet, in our method, we position the new point nearer to the minority cluster, reducing the impact of outliers. Within Fig. 5.4b, the initial blue value in parenthesis represents the distance from the median centroid. Meanwhile, the subsequent number exhibits the assigned weights for each neighbour, with those closer to the centroid receiving higher weights.

Algorithm 3 Distance ExtSMOTE

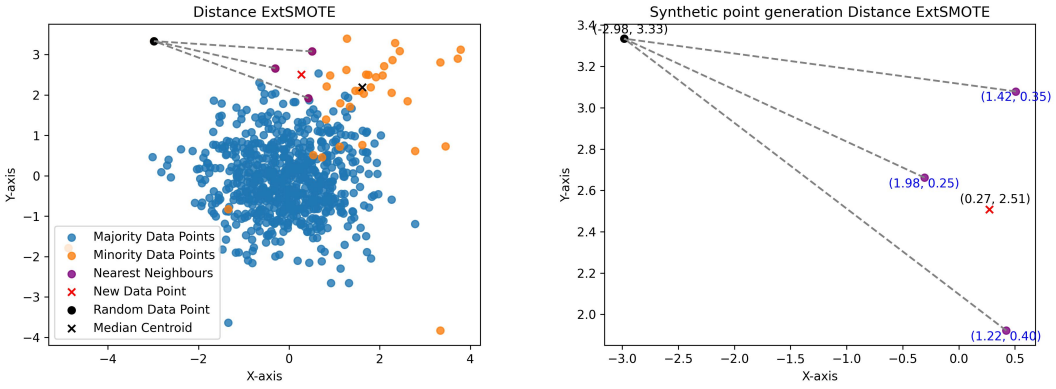
Require: $X \in \mathbb{R}^{n \times p}$ the features.

Require: $Y \in \{0, 1\}^n$ the binary class label outputs.

Require: $k \in \mathbb{N}$ the number of neighbors to select for the k-Nearest Neighbors.

Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{0, 1\}^q$ with q points created.

1: Denote by S_1 the number of points labelled as the minority class and S_0 the number of points labelled as the majority class.
2: Initialize X_{new} and Y_{new} as empty vectors.
3: Obtain the median centroid (μ) of the minority class.
4: **while** $S_1 < S_0$ **do**
5: Filter $\mathcal{D} = \{X_i | Y_i = 1\}$, the set of points labeled as minority class 1.
6: Randomly choose $r \in \mathcal{D}$ and find the indices of its k nearest neighbors, $r_1, \ldots, r_k$.
7: Consider the inverse distances, from μ, to each nearest neighbour as weights, $w_j = d_j^{-1}$
8: $x^{new} \leftarrow \frac{\sum (w_j \times x_{r_j})}{\sum w_j}$ for all j from 1 to k.
9: $y^{new} \leftarrow 1$
10: $S_1 = S_1 + 1$
11: Append x^{new} to X_{new}, append y^{new} to Y_{new}
12: return X_{new}, Y_{new}



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.

(b) The values within parentheses indicate (d_j, w_j).

Figure 5.4: Distance ExtSMOTE data generation

5.3.6 Dirichlet ExtSMOTE

The Dirichlet ExtSMOTE method introduces a probabilistic framework to assign weights. It utilizes the Dirichlet distribution parameters obtained through three distinct approaches. These parameters are then employed as weights for each nearest neighbour in the form of generated Dirichlet samples.

5.3.6.1 Dirichlet Distribution

The Dirichlet distribution is a multivariate probability distribution defined on the simplex, which allows us to capture the class proportions and guide the creation of more representative synthetic instances. The Dirichlet distribution is used in diverse applications across numerous domains. It plays a fundamental role in the modelling of compositional data, Bayesian analysis, statistical genetics, nonparametric inference, distribution-free tolerance intervals, multivariate analysis, order statistics, reliability, probability inequalities, probabilistic constrained programming models, limit laws, delivery problems, stochastic processes, and other areas (Ng et al.; 2011).

It also can be considered a distribution over probability distributions (Ng et al.; 2011; Bela et al.; 2010). Each draw from a Dirichlet distribution yields a vector of probabilities that can be interpreted as a probability distribution, where each element represents the likelihood of an outcome within a specific category or dimension.

Mathematically, the Dirichlet distribution is defined by a set of parameters $\alpha_1, \alpha_2, \ldots, \alpha_K$, where K is the dimensionality of the probability simplex. The parameters are positive real numbers, and the distribution is typically represented as $Dir(\boldsymbol{\alpha})$. The Dirichlet distribution is parameterized by $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_K]$, which can be considered pseudo-counts or prior observations.

Let $\boldsymbol{p} = [p_1, p_2, \ldots, p_k]$ be a K-dimensional vector such that for all $j = 1, \ldots, k$ we have $p_j \geq 0, j = 1, 2, \ldots, k$ and $\sum_{j=1}^{K} p_j = 1$. If Γ is the gamma function, then the probability density function (pdf) of the Dirichlet distribution for a point $\boldsymbol{p}$ on the simplex is given by (Bela et al.; 2010):

$$P(\boldsymbol{p}|\boldsymbol{\alpha}) \sim Dir(\alpha_1, \alpha_2, \ldots, \alpha_K) \stackrel{\text{def}}{=} \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{K} p_j^{\alpha_j - 1} \tag{5.2}$$

Three distinct approaches are utilized to generate Dirichlet parameters in the Dirichlet ExtSMOTE method. The first method involves generating random data from a $Uniform(0, 1)$ distribution, known as the uniform distribution approach. The second

method, termed the uniform vector approach, generates data from a unit vector of size k. Finally, the inverse distance approach determines parameters by considering the inverse distances between the median centroid and the chosen k nearest neighbors. To enhance variability with the concentration parameter, each vector is multiplied by a scalar m. These approaches collectively contribute to the diverse generation of Dirichlet parameters within the Dirichlet ExtSMOTE framework.

Algorithm 4 Dirichlet ExtSMOTE

Require: $X \in \mathbb{R}^{n \times p}$ the features.

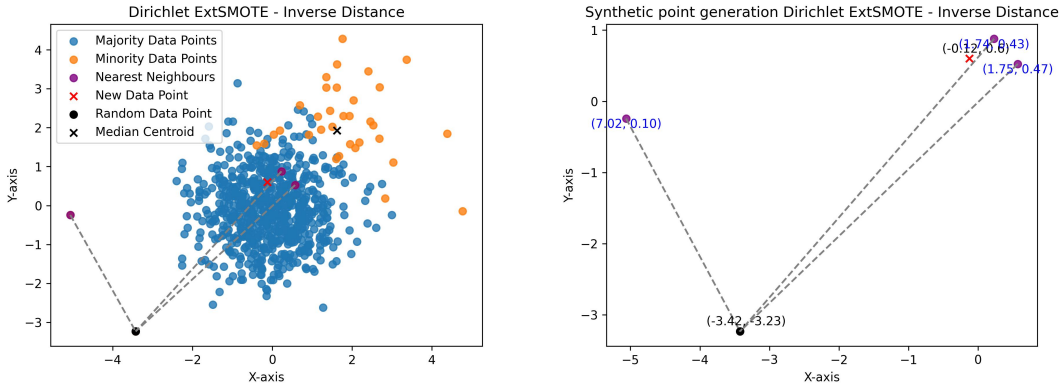
Require: $Y \in \{0,1\}^n$ the binary class label outputs.

Require: $k \in \mathbb{N}$ the number of neighbors to select for the k-Nearest Neighbors.

Require: $m > 0 \in \mathbb{R}$, the multiplier of the parameter of the distribution.

Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{0,1\}^q$ with q points created.

Denote by S_1 the number of points labelled as the minority class and S_0 the number of points labelled as the majority class.
Initialize X_{new} and Y_{new} as empty vectors.
Obtain the median centroid (μ) of the minority cluster.
while $S_1 < S_0$ **do**
    Filter $\mathcal{D} = \{X_i | Y_i = 1\}$, the set of points labeled as minority class 1.
    Randomly choose $r \in \mathcal{D}$ and find the indices of its k nearest neighbors, $r_1, \ldots, r_k$.
    if Type is 'Inverse distance (D)' **then**
        Calculate the distances, $\boldsymbol{D} = [d_1, \ldots, d_k]$ from μ to each nearest neighbour and obtain the reciprocal of each distance $\boldsymbol{D}^{-1} = [\frac{1}{d_1}, \ldots, \frac{1}{d_k}]$. Then $\boldsymbol{\alpha} = \boldsymbol{D}^{-1} \times m$
    else if Type is 'Uniform Vector (UV)' **then**
        Generate a vector $\boldsymbol{\alpha} = \mathbf{1_k} \times m$, where $\mathbf{1_k} = [1, \ldots, 1]$
    else if Type is 'Uniform Distribution (UD)' **then**
        Generate vector $\boldsymbol{U}$ of size k from $uniform(0,1)$ distribution, then $\boldsymbol{\alpha} = \boldsymbol{U} \times m$.
    Use $\boldsymbol{\alpha}$ as parameters to the Dirichlet Distribution and generate random weights $w_j \sim Dir(\boldsymbol{\alpha})$
    $x^{new} \leftarrow \sum w_j x_{r_j}$ for all j from 1 to k, as $\sum w_j = 1$
    $y^{new} \leftarrow 1$
    $S_1 = S_1 + 1$
    Append x^{new} to X_{new}, append y^{new} to Y_{new}
return X_{new}, Y_{new}



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.

(b) The values within parentheses indicate (d_j, w_j).

Figure 5.5: Dirichlet ExtSMOTE (Inverse Distance)

By incorporating Dirichlet weights, the algorithm promotes the creation of more diverse and relevant synthetic instances (Fig. 5.5). This results in synthetic samples that better reflect the characteristics of the minority class. The Dirichlet distribution's ability to generate diverse weights mitigates the risk of overfitting that traditional SMOTE might face.

The algorithm (Alg. 4) enables better generalization of classifiers on imbalanced datasets, improving performance in real-world applications with class imbalance challenges. In the subsequent section, we present the experimental setup and evaluation results, highlighting the advantages of Dirichlet ExtSMOTE over standard SMOTE and other state-of-the-art approaches in handling class imbalance.

5.3.7 FCRP SMOTE

The Chinese Restaurant Process (CRP) (Pitman and Picard; 2006) stands as a foundational probabilistic concept frequently employed to display random allocation or grouping in the domains of Bayesian statistics and machine learning. FCRP SMOTE adopts the CRP concept but integrates a predetermined number of tables, denoted as k, and assigns initial preferences to each table.

Consider a scenario with k nearest neighbours represented as tables in a restaurant setting. In this framework, customers emulate the role of developing the weights to aid in selecting a synthetic data point. Initially, there exist specific preferences allocated

to each table, which are essentially the normalized inverse distances. When a customer arrives, they select a table based on these probabilities. Subsequently, after this choice, the probability for that particular table is adjusted using an α value, and recalculations are made for all other probabilities in relation to that selection. This sequential process iterates for a total of N customers. The resulting probabilities obtained through these iterations serve as the desired weights for each nearest neighbour within the FCRP SMOTE methodology as shown in Alg. 5. Fig. 5.6 illustrates the generation of a synthetic instance.

Algorithm 5 FCRP SMOTE

Require: $X \in \mathbb{R}^{n \times m_1}$ the features, $Y \in \{0,1\}^n$ the binary class label outputs.

Require: $k \in \mathbb{N}$ the number of neighbors to select for the k-Nearest Neighbors.

Require: $\alpha \in \mathbb{R}$, scalar parameter to update preferences.

Ensure: Generated data $X_{new} \in \mathbb{R}^{m_2 \times m_1}$ and $Y_{new} \in \{0,1\}^{m_2}$ with m_2 number of points created.

Denote by S_1 the number of points labelled as the minority class and S_0 the number of points labelled as the majority class.

Initialize X_{new} and Y_{new} as empty vectors.

Filter $\mathcal{D} = \{X_i | Y_i = 1\}$, the set of points labeled as minority class 1 and obtain the median centroid (μ) of the minority cluster.

while $S1 < S_0$ **do**

    Randomly choose $r \in \mathcal{D}$ and find the indices of its k nearest neighbors, $\{r_1, \ldots, r_k\}$.

    Consider the normalized inverse distances, from μ, to each nearest neighbour as initial preferences, $P = \boldsymbol{D}_{norm}^{-1}$ and let the initial selection of the nearest neighbour occur with a probability p_i, i from $1, \ldots, k$.

    for N-1 **do**

        Choose the next nearest neighbour with the following updated probabilities q_i,

$$q_i = \begin{cases} \frac{p_i + \alpha}{1+\alpha}, & \text{for previously chosen neighbour} \\ \frac{p_i}{1+\alpha}, & \text{for other neighbours} \end{cases}$$

        $p_i = q_i$

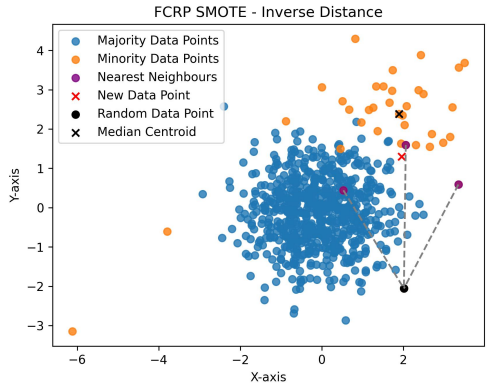
    Weights w_j are obtained from the final preferences for each neighbour p_i.

    $x^{new} \leftarrow \sum (w_j \times x_{r_j})$ for all j from 1 to k and $y^{new} \leftarrow 1$

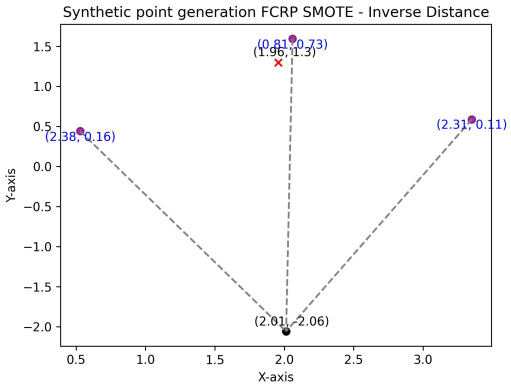
    $S_1 = S_1 + 1$

    Append x^{new} to X_{new}, append y^{new} to Y_{new}

return X_{new}, Y_{new}



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.



(b) The values within parentheses indicate (d_j, w_j).

Figure 5.6: FCRP SMOTE

5.3.8 BGMM SMOTE

In BGMM SMOTE (a cluster-based Synthetic Minority Over-sampling Technique), the approach relies on assigning each neighbouring data point to a specific cluster. Subsequently, the algorithm assigns weights to each nearest neighbour based on the likelihood associated with each cluster. This method uses the clustering information to guide the creation of synthetic instances in the minority class, considering the likelihood estimates derived from the clusters to generate new samples more effectively.

5.3.9 Bayesian Gaussian Mixture Models with EM Algorithm

In cases where abnormal instances are present within the minority class, it's often observed that these outlier points tend to reside at a considerable distance from the primary cluster. This also suggests that these points may belong to a distinct cluster separate from the main cluster where the median centroid resides. However, such disparity in data distribution raises a fundamental challenge: determining the number of clusters that emerge from these scattered data points. In such scenarios, traditional clustering methods may fall short due to the absence of prior knowledge about the data's inherent structure.

To address this challenge effectively, we turn to a non-parametric Bayesian approach called the Bayesian Finite Mixture Model (Roberts et al.; 1998). This approach allows

us to overcome the need to specify the number of clusters beforehand. Instead, it dynamically adapts to the data, flexibly accommodating the varying degrees of cluster complexity within the chosen dataset.

The likelihood function, given n independent and identically distributed observations $\boldsymbol{y_i} \in \mathbb{R}^p$ and a fixed number of components K, is defined as:

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}, K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{y_i}|\boldsymbol{\theta_k}) \tag{5.3}$$

where $\boldsymbol{y}$ are the observations from the mixture model, $f_k()$ a parametric density with parameters θ_k and π_k is the cluster mixing weight.

Furthermore, estimating cluster mixing weights, which define each cluster's relative contribution to the overall data distribution, becomes a critical aspect of this task. In the absence of prior information, Bayesian methods provide an appealing solution. By introducing a prior distribution over these cluster mixing weights, Bayesian inference is observed through the possibility of adapting the parameter K, which is, in practical use cases, often unknown.

The Expectation-Maximization (EM) algorithm is an iterative technique for fitting Gaussian Mixture Models. In the context of Bayesian Gaussian Mixture Models, the EM algorithm iterates between the Expectation (E) step and the Maximization (M) step:

1. E-step: Expectation

   Estimate the posterior probability of each data point belonging to each cluster, known as the responsibility, using the current parameters.

2. M-step: Maximization

   Update the parameters (such as means, covariances, and mixing weights) based on the newly calculated responsibilities.

The posterior distribution with a fixed K is represented as:

$$p(\theta|\boldsymbol{y}) \propto p(\theta) \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(y_i|\theta_k)$$

For the weight distribution, two distinct priors are integrated:

1. Dirichlet Distribution Prior

Uses a finite mixture model with a symmetric Dirichlet prior $Dir(e_0)$ for the component weights:

$$\pi_1, \ldots, \pi_K \sim Dir_K(e_0), \pi_k \geq 0 \text{ for all k and } \sum_{k=1}^{K} \pi_k = 1$$

2. Dirichlet Process Prior

   Uses an infinite mixture model through the Dirichlet Process, approximating the Dirichlet Process inference algorithm using a truncated distribution, known as the stick-breaking representation. This process allows for the flexible creation of an infinite number of components. Consider a stick length of 1, where $V_k \sim Beta(1, \alpha_0)$ for $k = 1, 2, 3, \ldots$. The probabilities π_i for each component are calculated by the length taken away in each step of this process (Blei and Jordan; 2006):

$$\pi_k(\boldsymbol{v}) = v_k \prod_{j=1}^{k-1} (1 - v_j)$$

   The stick-breaking process generates mixture weights from a distribution defined by the Dirichlet Process, allowing for flexibility in handling unknown or potentially infinite numbers of clusters in the data.

After obtaining the mixing weights, they are used for generating new synthetic data points, enriching the understanding of the data's underlying structure. In Bayesian Gaussian Mixture Models, cluster assignment is typically "soft," computing the probability of a point belonging to each cluster instead of definitively assigning it to a single cluster. This yields posterior probabilities $p(\theta|\boldsymbol{y})$, indicating the likelihood of individual data points belonging to each cluster within the mixture model.

Algorithm 6 BGMM SMOTE

Require: $X \in \mathbb{R}^{n \times p}$ the features.

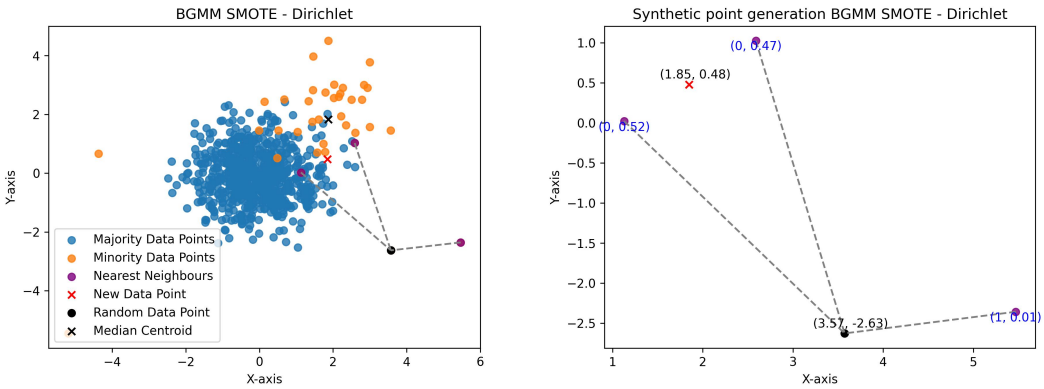
Require: $Y \in \{0, 1\}^n$ the binary class label outputs.

Require: $k \in \mathbb{N}$ the number of neighbors to select for the k-Nearest Neighbors.

Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{0, 1\}^q$ with q points created.

Denote by S_1 the number of points labelled as the minority class and S_0 the number of points labelled as the majority class.
Initialize X_{new} and Y_{new} as empty vectors.
Filter $\mathcal{D} = \{X_i | Y_i = 1\}$, the set of points labeled as minority class 1.
Obtain the median centroid (μ) of the minority cluster.
while $S_1 < S_0$ **do**
    Randomly choose $r \in \mathcal{D}$ and find the indices of its k nearest neighbors, $r_1, \ldots, r_k$.
    Estimate the mixing weights for each cluster for each selected data point, $r_1, \ldots, r_k, \mu$.
    if Prior is 'Dirichlet Distribution (D)' **then**
        Use Bayesian Gaussian Mixture with Dirichlet distribution as the weight concentration prior.
    else if Prior is 'Dirichlet Process (DP)' **then**
        Use Bayesian Gaussian Mixture with Dirichlet Process as the weight concentration prior.
    Retrieve the mixing weights w_j of the cluster to which the median centroid μ belongs.
    $x^{new} \leftarrow \frac{\sum (w_j \times x_{r_j})}{\sum w_j}$ for all j from 1 to k.
    $y^{new} \leftarrow 1$
    $S_1 = S_1 + 1$
    Append x^{new} to X_{new}, append y^{new} to Y_{new}
return X_{new}, Y_{new}

Fig. 5.7 demonstrates generating a new point using suggested BGMM SMOTE technique. Within Fig. 5.7b, the initial blue value in parenthesis represents the cluster to which the data point most likely belongs to. Meanwhile, the subsequent number exhibits the assigned weights for each neighbour, with those belonging to the same cluster as the centroid receiving higher weights.



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.

(b) The values within parentheses indicate (c_j, w_j).

Figure 5.7: BGMM SMOTE

5.3.10 Role of Weighting Mechanism in Mitigating Abnormal Instances

The weighting mechanism plays a critical role in reducing the influence of abnormal instances by addressing several key aspects, such as minimizing the influence of outliers, enhancing representative sampling and maintaining data integrity.

- In our methods, lower weights are assigned to such outliers due to their greater distance from the median centroid or lower probability of belonging to the core minority class distribution, ensuring they have minimal impact on generating synthetic samples while preserving the integrity and representativeness of the minority class data.
- By giving higher weights to instances closer to and more representative of the minority class, the synthetic samples generated are more reflective of the true distribution of the minority class. This leads to a more accurate and robust classification performance.
- The approach ensures that the synthetic data maintains the intrinsic properties of the minority class without being distorted by the extreme values introduced by abnormal instances. This is crucial for maintaining the validity and interpretability of the synthetic data.

- Abnormal instances are retained in our method because they may contain valuable information that can be useful to classification models. This retention ensures that potentially significant data is not lost during the resampling process.

5.3.11 Performance Measures

There are many widely accepted performance measures for binary classification; however, not all are suitable when dealing with imbalanced data. In our analysis, we used three key measures: F1 Score, PR-AUC, and MCC. Introducing the notations TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives),

Precision: $\frac{TP}{TP+FP}$, which measures the accuracy of the positive predictions, indicating the proportion of true positive instances among all instances predicted as positive.
Recall: $\frac{TP}{TP+FN}$, which measures the ability of the classifier to identify all positive instances, representing the proportion of true positive instances among all actual positive instances.
The selected measures are defined as follows:

- **F1 Score:** $2\times\frac{Precision\times Recall}{Precision+Recall}$, which is the harmonic mean of Precision and Recall, providing a single metric that balances the trade-off between them, especially useful when there is an uneven class distribution.
- **PR-AUC:** The area under the Precision-Recall curve, which summarizes the trade-off between Precision and Recall across different thresholds.
- **MCC (Matthews Correlation Coefficient):** $\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, which takes into account all four confusion matrix categories and provides a balanced measure, even for imbalanced datasets.

5.4 Simulation Results

5.4.1 Synthetic Data Generation

In our research, we performed an experimental study in which we created two distinct classes using a synthetic approach based on multivariate normal distributions. We generated data from two sets of bi-variate normal distributions, $\boldsymbol{X}_{minority-outliers} \sim \mathcal{N}_{2\times 2}(\boldsymbol{\mu}^{(1)}_{2\times 1}, \boldsymbol{\Sigma}^{(1)}_{2\times 2})$ and $\boldsymbol{X}_{majority} \sim \mathcal{N}_{2\times 2}(\boldsymbol{\mu}^{(2)}_{2\times 1}, \boldsymbol{\Sigma}^{(2)}_{2\times 2})$, representing the minority and majority classes, respectively. The simulation results in this manuscript are obtained using these distributions, including the means and covariances, which were specified as

follows:

$$\boldsymbol{\mu}^{(1)}_{2\times1} = \begin{bmatrix}0\\0\end{bmatrix}_{2\times1}, \boldsymbol{\Sigma}^{(1)}_{2\times2} = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}_{2\times2}, \boldsymbol{\mu}^{(2)}_{2\times1} = \begin{bmatrix}3\\4\end{bmatrix}_{2\times1}, \boldsymbol{\Sigma}^{(2)}_{2\times2} = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}_{2\times2}$$

To accentuate the challenges posed by abnormal instances within the minority class, we took a deliberate step to introduce data points from a uniform distribution as outliers, $\boldsymbol{X}_{outliers} \sim \text{Uniform}([-10, 10]^2)$. These abnormal instances were strategically added to the minority class to mimic scenarios where anomalous instances exist within the underrepresented class. Fig. 5.8 displays the bivariate plot of the resulting resampled data for one of these scenarios.

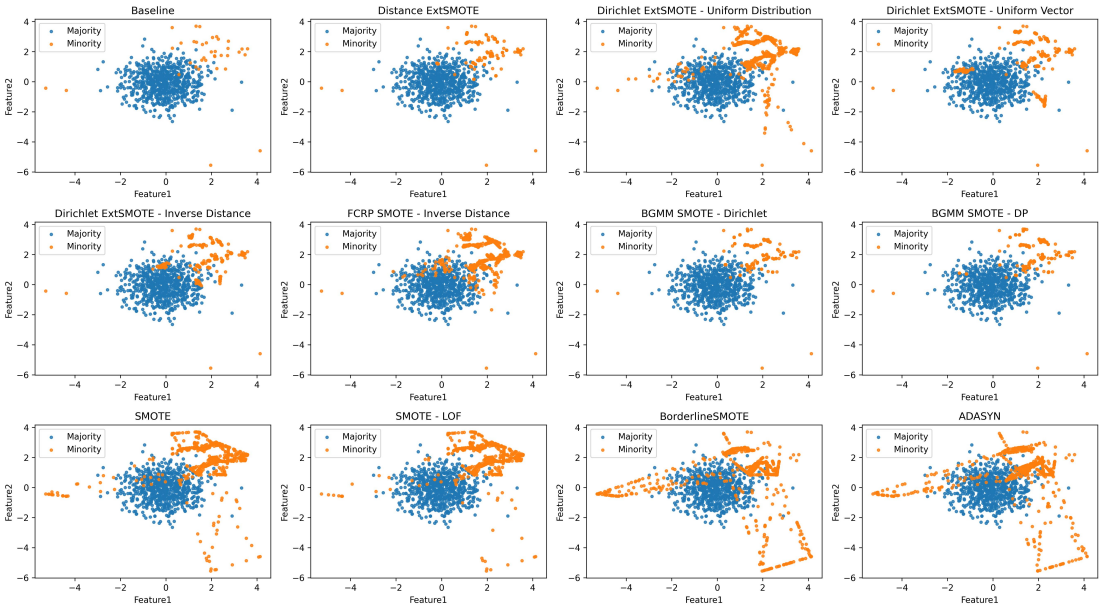


Figure 5.8: Results comparison of one simulated dataset with two features and five abnormal instances

To assess performance across various dataset shapes, we synthetically generated data using 'make_moons' and 'make_circles' libraries library in python scikit-learn.datasets (Pedregosa et al.; 2011). Figures in the appendix A.1 demonstrate the capacity of the proposed methods to generate synthetic data points, effectively mitigating the influence of abnormal instances. In contrast, SMOTE and its extensions form an inaccurate data bridge that links these abnormal instances with normal ones.

5.4.2 Simulation Setup

We generated a dataset comprising 1000 samples with two features, maintaining class imbalance ratios of 3 and 19. For an imbalance ratio of 3, the class distribution between majority and minority is 75:25, while for an imbalance ratio of 19, it is 95:5. We also considered two different outlier ratios, 0.05% and 0.1% in the analysis. Our initial step involved training classifiers without oversampling. This is the baseline method of our results. We used three different classifiers in the analysis: the Logistic Regression model, the K-Nearest Neighbors (KNN) classifier and the Random Forest classifier.

Subsequently, we trained the models with oversampled data generated by various proposed and several well-performed SMOTE variants, including BorderlineSMOTE, ADASYN, SMOTE-LOF (with negative outlier factor < -1.5 as outliers) (Asniar et al.; 2022) SMOTE IPF, DBSMOTE, ProWSyn, DSMOTE, SMOTE TomekLinks, SMOTE ENN, cluster SMOTE, Gaussian SMOTE. All algorithms, except for the proposed once and most common SMOTE, SMOTE-LOF, BorderlineSMOTE, and ADASYN, were implemented using the smote-variant python tool package (Kovács; 2019) available at `https://github.com/analyticalmindsltd/smote_variants`.

Then, oversampled data was applied only to the 75% training fold while the 25% test fold was used for validation. This process was repeated across 100 simulation datasets to mitigate variability and ensure robustness in our analysis.

In terms of parameterization, all sampling algorithms employ a shared neighbourhood parameter set at $k = 5$, ensuring an unbiased comparison. This particular setting at $k = 5$ is not only the most commonly used but also serves as the prevalent standard across SMOTE and most of its variants (Chawla et al.; 2002). For the classification models, the parameter k in KNN is also designated as five.

We also examined several key parameters to optimize the performance of our proposed methods. Specifically, we experimented with various values of alpha for FCRP SMOTE (0.1 and 0.5), different multipliers (0.01 and 100), types (distance, uniform distribution, uniform vector) for Dirichlet ExtSMOTE, and different priors (Dirichlet distribution, Dirichlet Process) for BGMM SMOTE. The optimal set of parameters, determined by the highest F1-score, was selected for each method, ensuring we identified the most effective configurations.

These final average F1-score, PR-AUC and MCC distributions when $OR \approx 0.05$ are graphically represented in Fig. 5.9, Fig. 5.10 and Fig. 5.11 respectively. The boxplots to the left of the dashed red line represent the proposed methods. The simulation setup demonstrates that the proposed methods outperform existing methods across different

machine learning classifiers, showcasing their robustness in classifier selection. Detailed results for $OR \approx 0.1$ for different IR, and accuracy measures are provided in the Appendix A.2. The results show similar results across these measures.

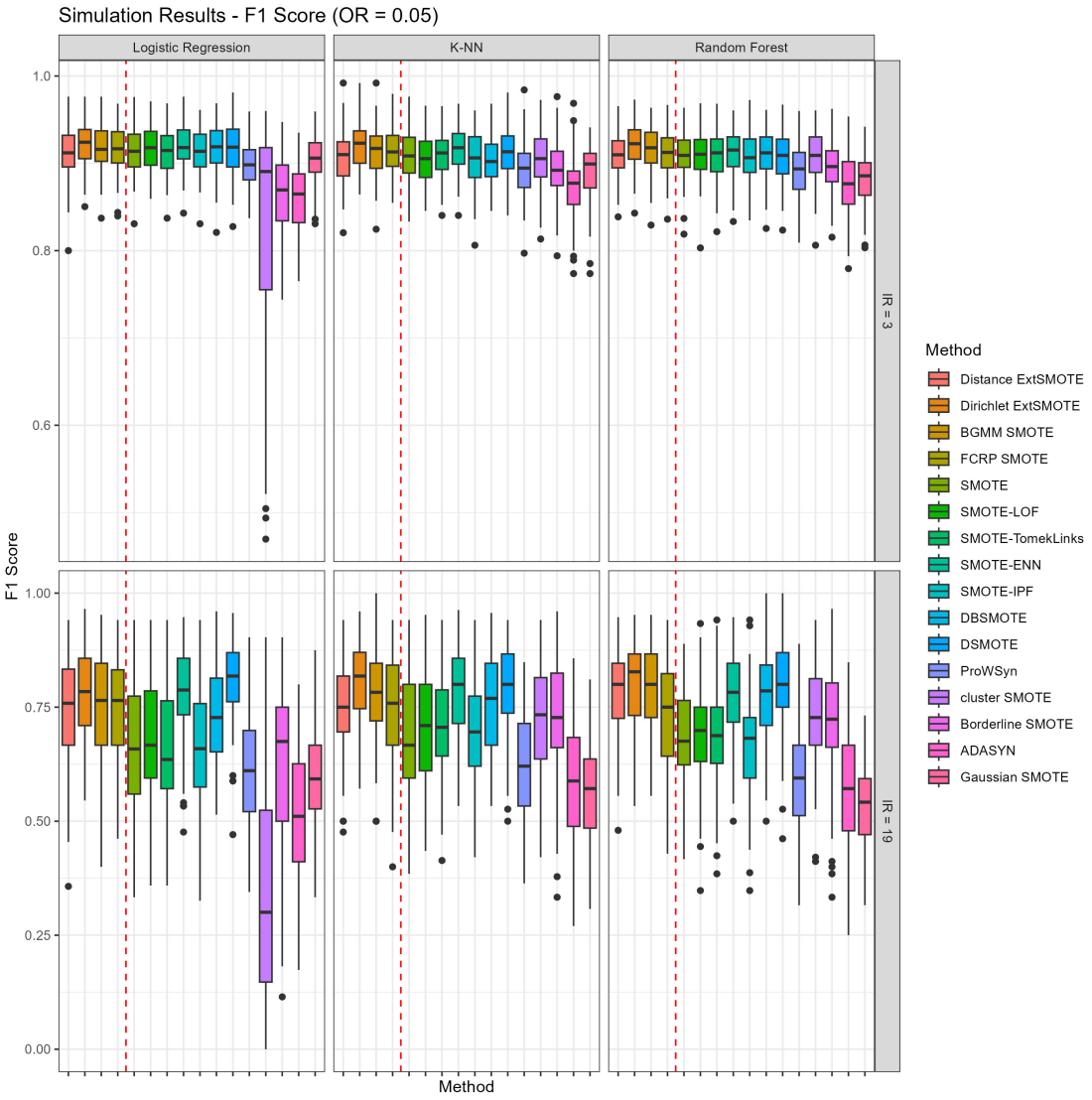


Figure 5.9: F1 Scores across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

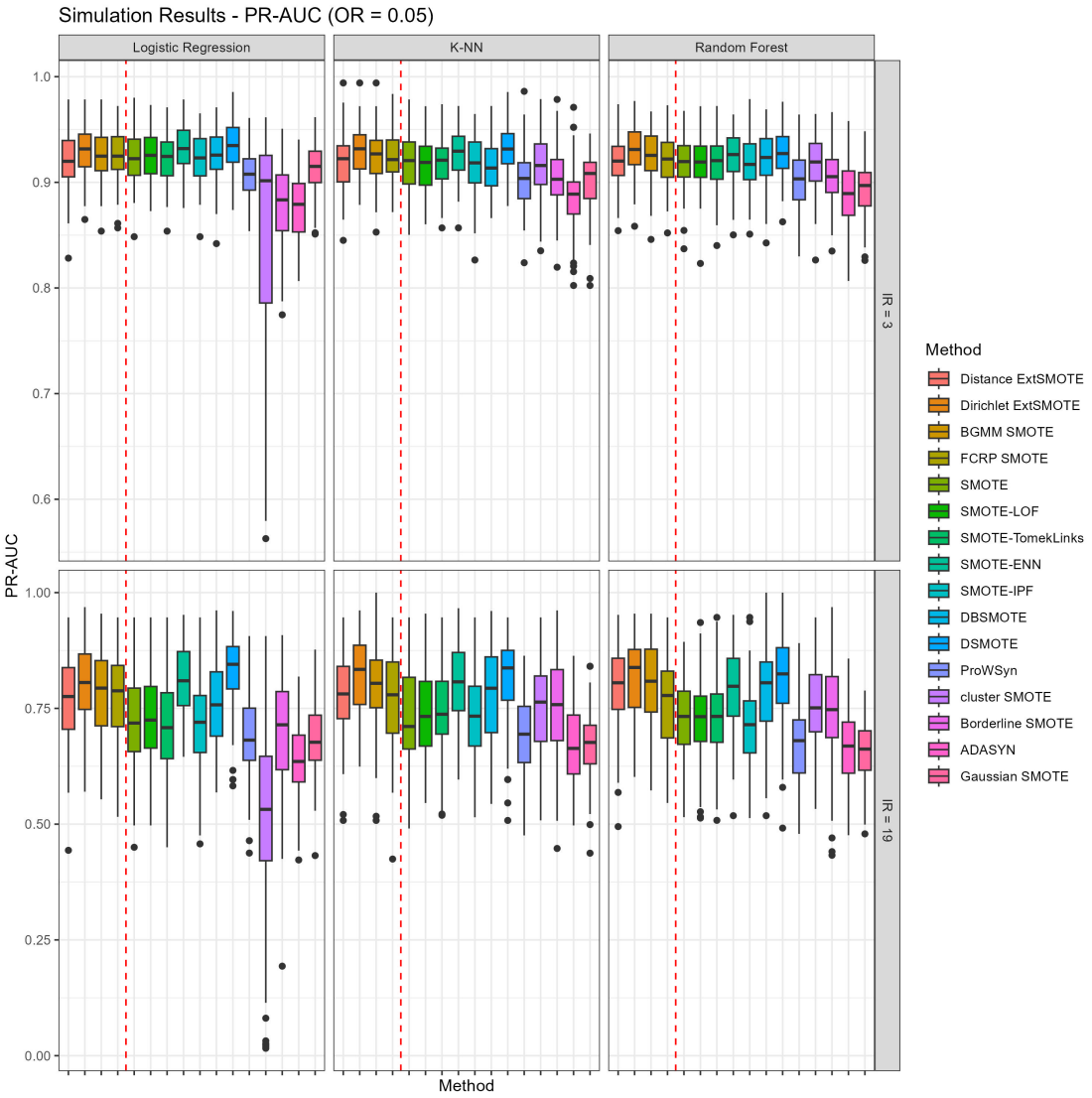


Figure 5.10: PR-AUCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

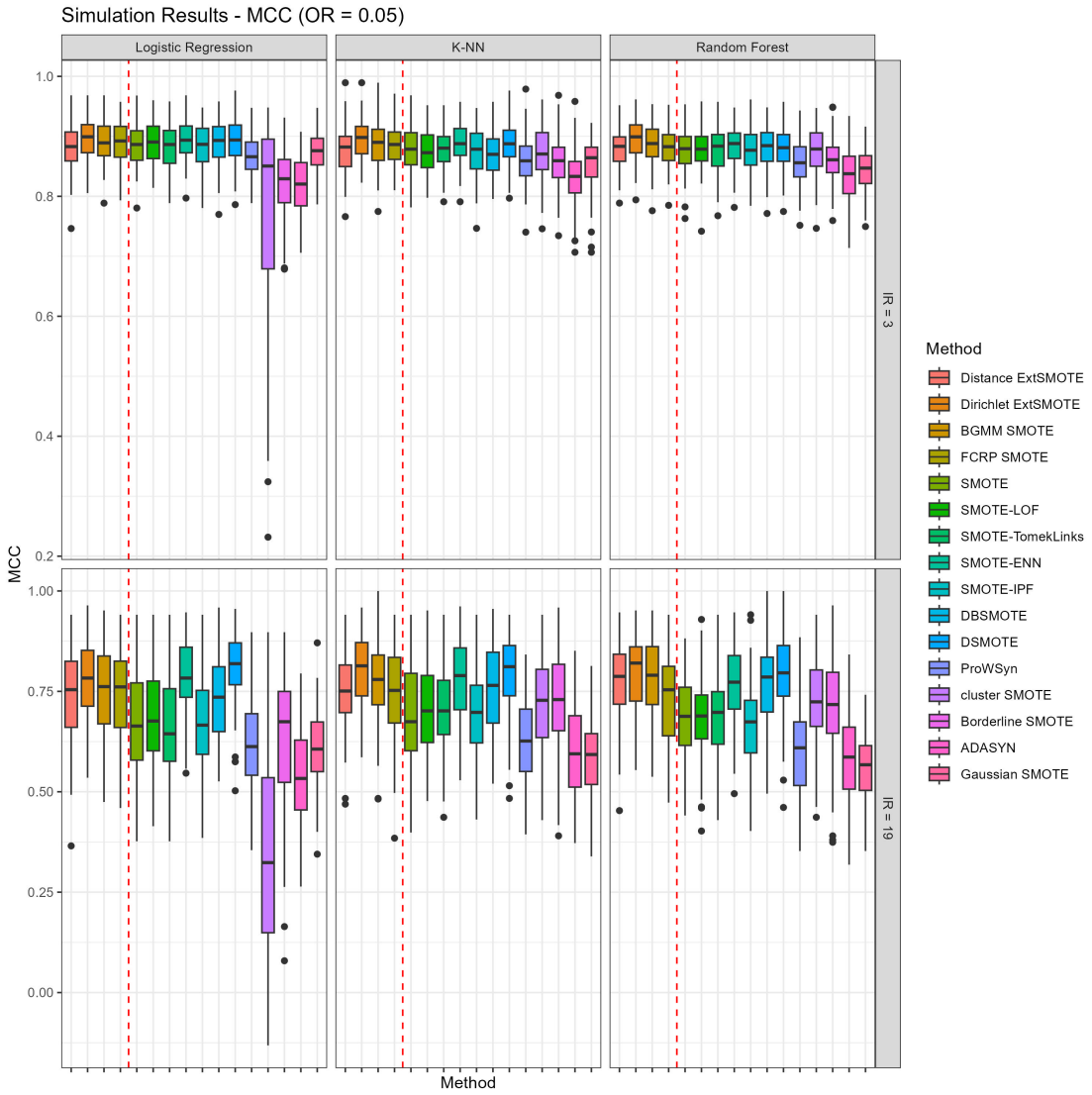


Figure 5.11: MCCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

5.4.3 Sensitivity Analysis of the Parameters

We systematically varied the parameter of interest for each proposed method while maintaining all other parameters constant. Specifically, for FCRP SMOTE, we tested various alpha values ranging from 0.1 to 1. For Dirichlet ExtSMOTE, we experimented with different multipliers (0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100) and types (inverse distance,

uniform distribution, uniform vector). For BGMM SMOTE, we explored different priors, including the Dirichlet distribution and the Dirichlet Process.

We then analyzed the impact of these changes on key performance metrics, including F1-score, PR-AUC, and MCC, using three different classifiers: Logistic Regression, KNN, and Random Forest. This comprehensive approach allowed us to enhance the performance of our proposed algorithms by identifying the most effective parameter configurations. The results are detailed in the Appendix A.3. The findings underscore the importance of carefully tuning parameters to achieve the best balance between model complexity and predictive accuracy in imbalanced datasets.

5.5 Application Results

5.5.1 Datasets Description

We collected 25 binary-class imbalanced data sets, commonly used in similar research scenarios, from the UCI machine learning repository (Newman and Merz; 1998; Pedregosa et al.; 2011) to verify the effectiveness of our proposed algorithm. These datasets have $3 - 100$ features, $159 - 6435$ instances, and the IR varies from 1.25 to 41.40. Detailed descriptions of the data sets are provided in Table 5.1. To understand the spatial distribution of anomalies in the dataset, we used the Local Outlier Factor (LOF) outlier detection method (Breunig et al.; 2000; Duan et al.; 2007). The figures in appendix A.4 show the principal component plots for the first two principal components, illustrating how the outliers are distributed within the minority class.

Table 5.1: Characteristics of the binary class datasets used in the computational study.

No	Dataset	Instances	Features	Minority class	Majority class	%Minority	%Majority	IR	Presence of LOF Outliers
1	yeast6 (Nakai; 1996)	1484	8	EXC	Remaining classes	2.36	97.64	41.40	Yes
2	yeast5 (Nakai; 1996)	1484	8	EXC, ERL	Remaining classes	2.70	97.30	36.10	Yes
3	yeast-1289vs7	947	8	VAC	NUC, CYT, ERL, POX	3.17	96.83	30.57	Yes
4	yeast4 (Nakai; 1996)	1484	8	ME2	Remaining classes	3.44	96.56	28.10	Yes
5	yeast-2vs8 (Nakai; 1996)	483	8	POX	CYT	4.14	95.86	23.15	Yes
6	glass12357vs6 (German; 1987)	214	9	6	Remaining classes	4.21	95.79	22.78	Yes
7	yeast-1458vs7 (Nakai; 1996)	693	8	VAC	NUC, ME3, ME2, POX	4.33	95.67	22.10	Yes
8	oil (Lemaitre et al.; 2017)	937	49	minority	majority	4.38	95.62	21.85	No
9	abalone9_18 (Nash et al.; 1995)	731	7	9, 18	Remaining classes	5.75	94.25	16.40	Yes
10	glass12367vs5 (German; 1987)	214	9	5	Remaining classes	6.07	93.93	15.46	Yes
11	thyroid_sick (Lemaitre et al.; 2017)	3772	52	sick	healthy	6.12	93.88	15.33	Yes
12	yeast-1vs7 (Nakai; 1996)	459	8	VAC	NUC	6.54	93.46	14.30	Yes
13	us_crime (Lemaitre et al.; 2017)	1994	100	>0.65	<=0.65	7.52	92.48	12.29	Yes
14	glass12vs5 (German; 1987)	159	9	5	1, 2	8.18	91.82	11.23	Yes
15	spectrometer (*Low Resolution Spectrometer*; 1988)	531	93	>=44	<44	8.47	91.53	10.80	Yes
16	landsat_satellite (Srinivasan; 1993)	6435	36	2	Remaining classes	9.73	90.27	9.28	Yes
17	mfeatmor0 (Duin; 2024)	2000	6	0, 1	Remaining classes	10.00	90.00	9.00	Yes
18	yeast3 (Nakai; 1996)	1484	8	ME3	Remaining classes	10.98	89.02	8.10	Yes
19	mfeatmor01 (Duin; 2024)	2000	6	0	Remaining classes	20.00	80.00	4.00	Yes
20	glass123vs567 (German; 1987)	214	9	5, 6, 7	Remaining classes	23.83	76.17	3.20	Yes
21	parkinsons (Little; 2008)	195	22	1	0	24.62	75.38	3.06	Yes
22	habermans_survival (Haberman; 1999)	306	3	2	1	26.47	73.53	2.78	Yes
23	glass23567vs1 (German; 1987)	214	9	1	Remaining classes	32.71	67.29	2.06	Yes
24	breast_cancer (Wolberg et al.; 1995)	569	30	M	B	37.26	62.74	1.68	Yes
25	banknote (Lohweg; 2013)	1372	4	1	Remaining classes	44.46	55.54	1.25	Yes

The proposed algorithms were also compared with the same SMOTE extensions used in the simulation study. These comparisons allowed us to evaluate the performance and robustness of our methods against established techniques under varying conditions.

5.5.2 Experimental Settings

In our experimental setup, we aimed to ensure robust model performance by incorporating comprehensive cross-validation and calibration steps. Cross-validation involved splitting the training data into five subsets, training the model on four subsets, and validating it on the remaining subset.

After the cross-validation step, we calibrated the classifiers using isotonic regression. Calibration was performed through the 'CalibratedClassifierCV' method in Python, which recalibrates the predicted probabilities to reflect the true likelihood of outcomes better. By integrating these steps, we ensured that the classifiers achieved high accuracy and provided well-calibrated probability estimates, crucial for reliable decision-making in imbalanced datasets.

Additionally, to ensure the robustness and generalizability of our findings, we conducted 100 iterations for each of the 25 application datasets. This extensive iteration process allowed us to account for variability in the results and provided a thorough evaluation of each method's performance across different datasets.

The rest of the experimental setup and parameter settings used in the application setup are consistent with those in the simulation study.

Moreover, in this study, we employed a ranking mechanism to evaluate and compare the performance of various classification methods based on their mean F1 scores across multiple classifiers, datasets, and trials. Specifically, for each unique combination of classifier, dataset, and trial, we calculated the mean F1 score for each method. Then, we ranked these methods in descending order, with higher mean F1 scores receiving better ranks. Ties were handled using the “min" method, which assigns the minimum rank to all tied values, ensuring that the ranks consistently reflect the best possible performance.

Fig. 5.12 shows the density plots of the rankings of each trial of different datasets for different re-sampling methods. The proposed methods are highlighted in red. The proposed density curves particularly Dirichlet SMOTE, FCRP SMOTE, and BGMM SMOTE, are skewed to the right regardless of the classifier, indicating that they achieved higher ranks (lower values) compared to existing methods.

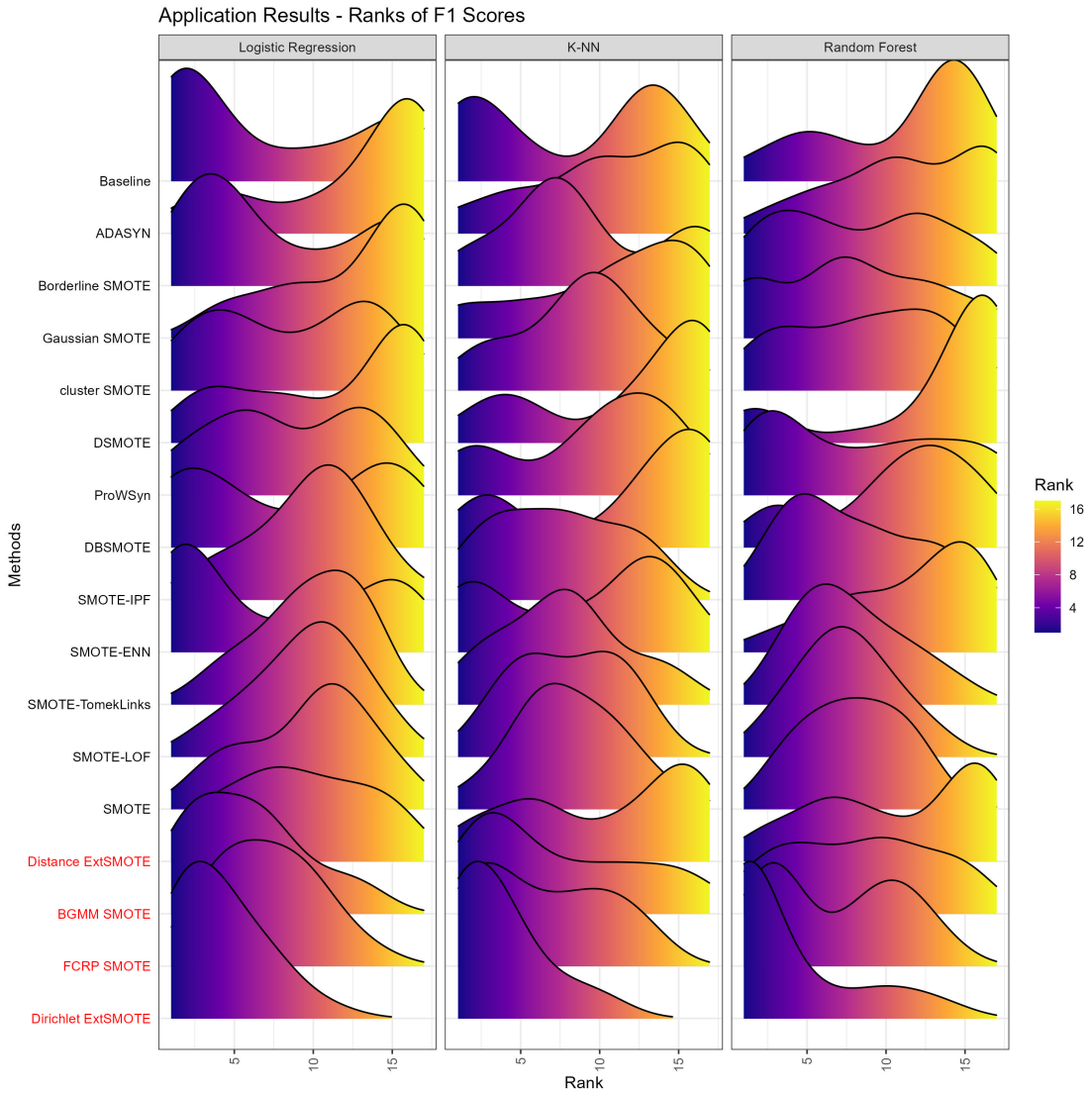


Figure 5.12: F1 Score Ranks for the datasets with 100 × 5-fold cross validation

5.5.2.1 Statistical Analysis

This ranking was further validated using the Friedman test for statistical significance (Friedman; 1937), and post-hoc pairwise comparisons were performed using Dunn's test with Holm adjustment (Dunn; 1964; Holm; 1979) to identify specific differences between methods, ultimately providing a robust comparative analysis of classification performance.

Tables 5.2, 5.3, and 5.4 present the test results for three classifiers, sorted according to average ranks. Dirichlet SMOTE consistently ranks first and rejects the null hypothesis

(H_0: no difference in the ranks of the groups being compared) every time, regardless of the classifier used. The Dirichlet SMOTE significantly differs from other SMOTE variants and suggested methods with all classifiers. This indicates that Dirichlet SMOTE is the most effective method, followed closely by FCRP SMOTE and BGMM SMOTE, showcasing their robustness across different classifiers. The fact that only the three proposed methods outperform the baseline highlights their superiority in handling class imbalance.

Table 5.2: Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on Logistic classifier.

	Method	Average_Rank	Distance ExtSMOTE	Dirichlet ExtSMOTE	FCRP SMOTE	BGMM SMOTE
1	Dirichlet ExtSMOTE	4.49	Rejected	-	Rejected	Rejected
2	BGMM SMOTE	6.37	Rejected	Rejected	Rejected	-
3	FCRP SMOTE	7.15	Rejected	Rejected	-	Rejected
4	Baseline	7.40	Rejected	Rejected	Not Rejected	Rejected
5	SMOTE-ENN	7.64	Rejected	Rejected	Not Rejected	Rejected
6	Borderline SMOTE	7.75	Rejected	Rejected	Rejected	Rejected
7	DBSMOTE	8.46	Not Rejected	Rejected	Rejected	Rejected
8	ProWSyn	8.60	Not Rejected	Rejected	Rejected	Rejected
9	Distance ExtSMOTE	8.65	-	Rejected	Rejected	Rejected
10	cluster SMOTE	8.93	Not Rejected	Rejected	Rejected	Rejected
11	SMOTE-LOF	9.16	Rejected	Rejected	Rejected	Rejected
12	SMOTE-IPF	9.31	Rejected	Rejected	Rejected	Rejected
13	SMOTE-TomekLinks	9.34	Rejected	Rejected	Rejected	Rejected
14	SMOTE	9.40	Rejected	Rejected	Rejected	Rejected
15	DSMOTE	10.94	Rejected	Rejected	Rejected	Rejected
16	ADASYN	12.01	Rejected	Rejected	Rejected	Rejected
17	Gaussian SMOTE	12.42	Rejected	Rejected	Rejected	Rejected

Table 5.3: Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on KNN classifier.

	Method	Average_Rank	Distance ExtSMOTE	Dirichlet ExtSMOTE	FCRP SMOTE	BGMM SMOTE
1	Dirichlet ExtSMOTE	4.43	Rejected	-	Rejected	Rejected
2	FCRP SMOTE	6.45	Rejected	Rejected	-	Rejected
3	BGMM SMOTE	7.37	Rejected	Rejected	Rejected	-
4	SMOTE-LOF	7.59	Rejected	Rejected	Rejected	Not Rejected
5	SMOTE-IPF	7.64	Rejected	Rejected	Rejected	Not Rejected
6	SMOTE-TomekLinks	7.71	Rejected	Rejected	Rejected	Not Rejected
7	SMOTE	7.81	Rejected	Rejected	Rejected	Not Rejected
8	Baseline	8.27	Rejected	Rejected	Rejected	Rejected
9	Borderline SMOTE	8.48	Rejected	Rejected	Rejected	Rejected
10	SMOTE-ENN	8.71	Rejected	Rejected	Rejected	Rejected
11	cluster SMOTE	8.75	Rejected	Rejected	Rejected	Rejected
12	ProWSyn	8.93	Rejected	Rejected	Rejected	Rejected
13	Distance ExtSMOTE	10.11	-	Rejected	Rejected	Rejected
14	ADASYN	10.19	Not Rejected	Rejected	Rejected	Rejected
15	Gaussian SMOTE	10.42	Not Rejected	Rejected	Rejected	Rejected
16	DBSMOTE	11.33	Rejected	Rejected	Rejected	Rejected
17	DSMOTE	11.57	Rejected	Rejected	Rejected	Rejected

Table 5.4: Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on Random Forest classifier.

	Method	Average_Rank	Distance ExtSMOTE	Dirichlet ExtSMOTE	FCRP SMOTE	BGMM SMOTE
1	Dirichlet ExtSMOTE	4.55	Rejected	-	Rejected	Rejected
2	FCRP SMOTE	6.93	Rejected	Rejected	-	Rejected
3	Gaussian SMOTE	7.82	Rejected	Rejected	Rejected	Rejected
4	SMOTE-LOF	7.86	Rejected	Rejected	Rejected	Rejected
5	ProWSyn	8.09	Rejected	Rejected	Rejected	Not Rejected
6	SMOTE-IPF	8.19	Rejected	Rejected	Rejected	Not Rejected
7	SMOTE-TomekLinks	8.21	Rejected	Rejected	Rejected	Not Rejected
8	SMOTE	8.25	Rejected	Rejected	Rejected	Not Rejected
9	Borderline SMOTE	8.27	Rejected	Rejected	Rejected	Not Rejected
10	BGMM SMOTE	8.38	Rejected	Rejected	Rejected	-
11	cluster SMOTE	8.56	Rejected	Rejected	Rejected	Not Rejected
12	Baseline	9.75	Rejected	Rejected	Rejected	Rejected
13	DBSMOTE	10.27	Not Rejected	Rejected	Rejected	Rejected
14	SMOTE-ENN	10.33	Not Rejected	Rejected	Rejected	Rejected
15	Distance ExtSMOTE	10.41	-	Rejected	Rejected	Rejected
16	ADASYN	10.57	Not Rejected	Rejected	Rejected	Rejected
17	DSMOTE	12.15	Rejected	Rejected	Rejected	Rejected

For the same experiment, Table 5.5, Table 5.6, and Table 5.7 compare the average F1 scores obtained by each resampling method and the baseline dataset. Results for other performance measures, PR-AUC and MCC are presented in the appendix A.5. The

tables highlight the highest F1 score for each dataset in bold, indicating that Dirichlet SMOTE consistently delivers superior performance across most datasets, particularly with the Random Forest classifier. ADASYN, however, may exhibit inherent limitations in handling certain data distributions or noise levels, as observed in the banknote dataset.

Table 5.5: F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100×5 fold cross-validation with a Logistic Regression classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.3812	0.3843	0.4213	0.3634	0.3065	0.3078	0.3057	**0.4315**	0.306	0.4157	0.2939	0.3235	0.2381	0.257	0.4192	0.2579	**0.4315**
yeast5	0.4065	0.4064	0.4311	0.3924	0.3637	0.3601	0.3589	0.48	0.3583	**0.5557**	0.3336	0.4354	0.3211	0.3128	0.4382	0.2997	0.48
yeast-1289vs7	0.1797	**0.2286**	0.1918	0.1738	0.1758	0.1754	0.1663	0.2071	0.1651	0.0939	0.1618	0.2079	0.2184	0.1979	0.2145	0.1645	0.2071
yeast4	0.2423	0.2858	0.3062	0.265	0.2571	0.2598	0.2604	0.2584	0.2625	**0.382**	0.2589	0.2478	0.2658	0.2616	0.3382	0.2399	0.2584
yeast-2vs8	0.2189	0.3927	0.2614	0.2517	0.4727	0.4704	0.448	0.5733	0.4454	0.2802	0.4979	0.6133	**0.6169**	0.4999	0.2075	0.1347	0.5733
glass12357vs6	0.4901	0.6299	0.5467	0.5704	0.6133	0.6133	0.6133	0.48	0.6133	**0.6489**	0.6133	0.4218	0.6079	0.4329	0.6133	0.6133	0.48
yeast-1458vs7	0.1656	**0.1978**	0.1685	0.1816	0.1696	0.1689	0.1653	0	0.1631	0.1256	0.1392	0.1592	0.1361	0.1765	0.1777	0.164	0
oil	0.5596	0.5582	0.5141	0.5197	0.4951	0.4954	0.5009	**0.5598**	0.5001	0.5046	0.5082	0.4509	0.4875	0.3919	0.482	0.4936	**0.5598**
abalone9_18	0.5101	0.5089	0.4759	0.4841	0.4245	0.4276	0.4427	0.4733	0.4426	**0.5401**	0.4785	0.4739	0.4609	0.3753	0.5271	0.3908	0.4733
glass12367vs5	0.4654	0.5624	0.5612	0.5015	0.5364	0.5366	0.5306	0.3733	0.5324	0.2267	0.5504	0.3438	0.4302	0.4733	**0.574**	0.5454	0.3733
thyroid_sick	0.5922	0.5995	0.5655	0.5814	0.5567	0.5611	0.5539	0.7067	0.5531	**0.7103**	0.5858	0.5894	0.6357	0.3493	0.5318	0.5198	0.7067
yeast-1vs7	0.3625	0.3536	**0.3747**	0.3435	0.2995	0.3001	0.303	0.1833	0.3041	0.2636	0.3019	0.3102	0.334	0.3457	0.3337	0.2969	0.1833
us_crime	0.4902	0.4986	0.5016	0.5014	0.4841	0.4832	0.48	0.4924	0.4815	0.4791	0.5	0.5171	0.5016	0.4764	**0.53**	0.4679	0.4924
glass12vs5	0.6381	0.7966	0.6984	0.779	0.7887	0.7874	0.7845	0.7381	0.7845	0.5017	0.7969	0.6219	0.7582	0.6566	0.7511	**0.8005**	0.7381
spectrometer	0.8538	0.857	0.8618	0.8549	0.8473	0.8447	0.8493	0.8617	0.8492	0.8472	0.8575	0.8284	0.8595	0.7929	0.8514	0.8334	**0.8676**
landsat_satellite	0.2907	0.2929	0.2923	0.2919	0.2941	0.2935	0.2936	0.0453	0.2941	0.0528	**0.2978**	0.1563	0.2961	0.2745	0.2934	0.2894	0.0453
mfeatmor0	0.9923	0.9931	0.9923	0.9923	0.9916	0.9923	0.9911	0.9909	0.9914	0.9923	**0.9945**	0.9731	0.9871	0.9764	0.9923	0.9899	0.9923
yeast3	0.7032	0.6992	0.703	0.6936	0.6425	0.6434	0.6445	**0.7582**	0.6446	0.7106	0.6695	0.7302	0.6445	0.6492	0.6409	0.6323	**0.7582**
mfeatmor01	0.9172	0.9174	0.9132	0.9157	0.9078	0.9114	0.9116	0.9239	0.9079	0.9238	0.907	0.9059	0.9062	0.8624	0.8335	0.7638	**0.9258**
glass123vs567	0.8303	0.8573	**0.8657**	0.857	0.8434	0.8421	0.8445	0.7932	0.8449	0.8401	0.8334	0.8373	0.8432	0.8315	0.8345	0.8391	0.7902
parkinsons	0.6616	**0.6794**	0.6677	0.6658	0.658	0.6596	0.6569	0.6752	0.6591	0.6739	0.6426	0.6344	0.6565	0.6192	0.6677	0.6678	0.6752
habermans_survival	0.4874	**0.5196**	0.5028	0.5017	0.4981	0.5	0.4996	0.3157	0.5013	0.4924	0.5008	0.3419	0.515	0.4777	0.4859	0.5097	0.3157
glass23567vs1	0.6362	0.6494	0.6423	0.6385	0.6235	0.6247	0.6269	**0.6594**	0.6222	0.625	0.6323	0.6466	0.6406	0.6399	0.6462	0.642	**0.6594**
breast_cancer	0.9643	0.9661	0.9663	0.9647	0.9629	0.9627	0.9617	**0.9664**	0.9634	0.9646	0.9612	0.9318	0.9624	0.9631	0.9505	0.9501	0.9653
banknote	0.9888	**0.9903**	0.9902	0.9897	0.9892	0.9891	0.9892	0.9892	0.9891	0.9889	0.9894	0.9412	0.9892	0.972	0.9892	NA	0.9892

Table 5.6: F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100×5 fold cross-validation with a K-NN classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.4724	0.5098	**0.5153**	0.5001	0.4047	0.4046	0.3992	0.4978	0.4024	0.4454	0.3615	0.4614	0.4057	0.3628	0.3985	0.362	0.4978
yeast5	0.5118	0.6008	0.5543	0.574	0.4949	0.4914	0.4845	**0.6487**	0.4891	0.6413	0.4287	0.5988	0.5135	0.3931	0.5397	0.4394	**0.6487**
yeast-1289vs7	0.1364	0.2636	0.2465	0.2231	0.2402	0.2452	0.2379	0.1943	0.2401	0.1016	0.1892	**0.3036**	0.2699	0.232	0.2406	0.2379	0.1943
yeast4	0.3266	**0.4089**	0.3771	0.3882	0.3557	0.3543	0.3662	0.2914	0.3608	0.3051	0.3177	0.3479	0.351	0.3713	0.3631	0.349	0.2914
yeast-2vs8	0.48	0.5513	0.4342	0.4188	0.4008	0.4004	0.3755	**0.6033**	0.3794	0.58	0.4085	0.5783	0.4442	0.413	0.3476	0.3704	**0.6033**
glass12357vs6	**0.8**	0.7783	0.658	0.7483	0.7833	0.7863	0.7917	0.7333	0.7927	0.7333	0.726	0.6811	0.7353	0.5037	0.783	0.7833	0.7333
yeast-1458vs7	0.0333	**0.1732**	0.0494	0.0841	0.1403	0.1403	0.1426	0.04	0.1472	0.1071	0.1591	0.1204	0.1545	0.1238	0.1399	0.1326	0.04
oil	0.4897	0.6003	0.5557	0.563	0.5698	0.5725	0.5722	0.5463	0.5732	0.5927	0.5659	0.3487	0.5536	0.4232	**0.6147**	0.5614	0.5463
abalone9_18	0.3519	0.4253	**0.4507**	0.4432	0.4044	0.4032	0.4089	0.4205	0.4089	0.4025	0.4058	0.4292	0.3629	0.4027	0.4185	0.3804	0.4205
glass12367vs5	0.5733	0.6712	0.633	0.6707	0.6631	0.6674	0.6823	0.4171	**0.6926**	0.2	0.661	0.3777	0.5714	0.5552	0.6508	0.6387	0.4171
thyroid_sick	0.5249	0.5566	0.5612	0.568	0.5779	0.575	0.5737	0.5472	0.5775	0.4542	0.5461	0.4904	0.5686	0.4978	0.56	**0.58**	0.5472
yeast-1vs7	0.3249	0.4092	0.3348	0.3822	0.4041	0.4038	0.3936	**0.4537**	0.3937	0.2564	0.3545	0.3693	0.4267	0.3583	0.4304	0.3871	**0.4537**
us_crime	0.3877	0.4125	0.4125	0.4186	0.4059	0.4054	0.4091	0.4025	0.4048	0.3405	**0.4332**	0.4135	0.4094	0.4096	0.4065	0.3948	0.4025
glass12vs5	0.28	0.576	0.5293	0.5582	0.6518	0.6347	0.6679	0.3314	0.6643	0.2133	0.6611	0.2976	0.6165	**0.6892**	0.6443	0.6577	0.3314
spectrometer	0.8018	0.8747	0.7645	0.8329	0.8416	0.8461	0.8492	0.836	0.8469	0.8107	0.824	0.8039	0.8242	**0.8754**	0.7935	0.8223	0.8478
landsat_satellite	0.6546	**0.7056**	0.6607	0.6837	0.7011	0.7011	0.6969	0.7029	0.6979	0.7012	0.6865	0.515	0.6925	0.6032	0.6871	0.6946	0.7029
mfeatmor0	0.9923	**0.9933**	0.9923	0.9923	0.9847	0.9888	0.986	0.9899	0.9868	0.9923	0.9885	0.9498	0.9876	0.9646	0.9923	0.9875	0.9923
yeast3	0.714	0.7359	0.7478	0.7421	0.7116	0.712	0.7168	0.7572	0.7147	0.6762	**0.7589**	0.7428	0.6825	0.7151	0.6325	0.6865	0.7572
mfeatmor01	0.9488	0.9501	0.9505	0.9479	0.9453	0.9452	0.9462	0.9509	0.9461	0.9375	0.9453	0.9357	0.9418	0.9422	0.919	0.8869	**0.9559**
glass123vs567	0.8628	**0.8753**	0.8704	0.8716	0.8542	0.8528	0.8552	0.8561	0.8523	0.8512	0.8709	0.8253	0.8479	0.8703	0.843	0.8441	0.8578
parkinsons	0.8586	0.8601	0.8446	0.8566	0.8283	0.8304	0.8278	0.7668	0.8344	0.7293	0.8062	0.7379	0.8318	0.8382	0.8493	**0.8645**	0.7668
habermans_survival	0.2906	0.4161	0.3847	0.3847	0.4123	0.4118	0.4054	0.1628	0.4145	0.2838	0.4458	0.3086	0.3882	**0.455**	0.4118	0.4159	0.1628
glass23567vs1	0.7238	**0.7536**	0.7348	0.7375	0.714	0.7136	0.7146	0.6828	0.7127	0.6651	0.6875	0.6636	0.7361	0.6807	0.7195	0.7283	0.6828
breast_cancer	0.947	**0.9544**	0.9493	0.9487	0.9473	0.948	0.9455	0.9438	0.948	0.9432	0.9453	0.9219	0.9455	0.9439	0.9382	0.9385	0.9434
banknote	0.9984	**0.9987**	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9982	0.947	0.9984	0.9859	0.9984	NA	0.9984

Table 5.7: F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5382	0.6059	**0.6232**	0.6024	0.5449	0.5516	0.5456	0.5086	0.5519	0.5124	0.4946	0.5107	0.5656	0.539	0.5272	0.5301	0.5077
yeast5	0.6019	0.6213	0.6171	0.6249	0.5287	0.533	0.5245	0.5959	0.5197	0.5845	0.5095	**0.6442**	0.5513	0.6196	0.529	0.5123	0.5969
yeast-1289vs7	0.0643	0.147	0.0929	0.1196	0.1771	0.1741	0.1847	0.2651	0.1751	0.1012	0.1523	0.2933	0.2773	**0.3319**	0.2354	0.1451	0.2632
yeast4	0.262	0.3913	0.3735	0.3816	0.4099	0.4081	0.3973	0.327	0.3923	0.3319	0.3812	0.3293	0.4313	0.3941	**0.4374**	0.4006	0.3307
yeast-2vs8	0.5393	0.5573	0.5014	0.4629	0.4297	0.4273	0.4202	0.5599	0.4215	0.5823	0.4859	0.5704	0.511	**0.6173**	0.338	0.2939	0.5601
glass12357vs6	0.6933	**0.7154**	0.6453	0.6329	0.6067	0.6127	0.6067	0.56	0.5993	0.6033	0.6253	0.5253	0.6107	0.6105	0.612	0.5887	0.565
yeast-1458vs7	0.0023	**0.1624**	0.043	0.091	0.1335	0.1279	0.1296	0.0021	0.1218	0.0026	0.1376	0.1246	0.0997	0.0441	0.1396	0.1253	0.0048
oil	0.4213	0.5253	0.4851	0.55	0.4825	0.4881	0.4949	0.4496	0.4952	0.2503	**0.5758**	0.4259	0.427	0.5543	0.531	0.4684	0.4538
abalone9_18	0.3747	**0.4172**	0.3851	0.4	0.3711	0.3707	0.3774	0.3634	0.3755	0.3975	0.3962	0.3744	0.3601	0.3823	0.3963	0.3567	0.3692
glass12367vs5	0.6526	**0.8326**	0.6594	0.7103	0.7207	0.7191	0.7339	0.5929	0.7345	0.7129	0.7586	0.6097	0.6606	0.7248	0.7426	0.7187	0.591
thyroid_sick	0.858	**0.8835**	0.8687	0.8587	0.8749	0.8704	0.8722	0.8592	0.8728	0.8551	0.8489	0.8127	0.8778	0.8694	0.8683	0.8711	0.8582
yeast-1vs7	0.1601	0.3189	0.2205	0.2946	0.3581	0.3563	0.3413	0.3519	0.3542	0.1272	0.2641	**0.4023**	0.3603	0.3787	0.3596	0.3132	0.3512
us_crime	0.4585	0.48	0.4607	0.4876	0.4986	0.4986	0.4958	0.4899	0.4992	0.3462	0.5187	0.4831	0.4861	**0.5229**	0.4981	0.4882	0.4932
glass12vs5	0.676	**0.8031**	0.7708	0.7831	0.7842	0.7916	0.789	0.6836	0.7862	0.7435	0.7885	0.5678	0.7306	0.782	0.785	0.7821	0.6821
spectrometer	0.8233	**0.8501**	0.8066	0.8283	0.8127	0.82	0.8144	0.8359	0.8128	0.8303	0.8084	0.8053	0.8116	0.82	0.8126	0.7978	0.8236
landsat_satellite	0.6514	0.6746	0.6696	0.6712	0.6828	0.6791	0.6812	0.6642	0.6812	0.6628	0.6798	0.6492	**0.6867**	0.6384	0.6836	0.6835	0.6641
mfeatmor0	0.9923	**0.9925**	0.9923	0.9923	0.9889	0.9914	0.9887	0.9891	0.9883	0.9909	0.9912	0.9528	0.9877	0.9879	0.9923	0.9871	0.9923
yeast3	0.7656	0.7866	0.7776	0.782	0.7839	0.7852	0.7845	0.759	0.7852	0.7678	**0.7867**	0.7521	0.783	0.7684	0.7776	0.7758	0.7602
mfeatmor01	0.9527	0.9579	0.9559	0.9522	0.9504	0.9511	0.9501	0.9533	0.95	0.9545	0.9568	0.9363	0.9484	0.9538	0.9317	0.891	**0.9581**
glass123vs567	0.8906	**0.9037**	0.8905	0.8998	0.8883	0.8882	0.8869	0.8859	0.8939	0.8871	0.8963	0.8191	0.8877	0.8898	0.8772	0.8764	0.8854
parkinsons	0.7659	**0.8106**	0.7976	0.8013	0.8053	0.8089	0.8093	0.7833	0.8104	0.7956	0.787	0.7176	0.8082	0.7904	0.7982	0.8069	0.7824
habermans_survival	0.3903	0.4228	0.4006	0.4283	0.3855	0.3825	0.4044	0.322	0.4089	0.3415	0.4321	0.3177	0.3793	**0.4566**	0.3762	0.3837	0.3197
glass23567vs1	0.8147	**0.8457**	0.8274	0.8234	0.8177	0.818	0.8152	0.8179	0.814	0.8262	0.7899	0.8124	0.8165	0.7873	0.814	0.8107	0.8223
breast_cancer	0.9474	**0.9521**	0.9481	0.9493	0.9468	0.947	0.9483	0.9445	0.947	0.9471	0.946	0.9219	0.9472	0.9459	0.944	0.9497	0.9447
banknote	0.9929	**0.994**	0.9929	0.9935	0.9926	0.9925	0.9925	0.9922	0.9926	0.9924	0.9925	0.938	0.9926	0.9901	0.9922	NA	0.9922

5.6 Discussion

Class imbalance is a common challenge in many real-world datasets, where one class is significantly underrepresented compared to the other(s). In such situations, it can be beneficial to generate synthetic data that addresses the class imbalance and minimizes the impact of abnormal instances, especially within the minority class. Abnormal instances in the minority class can introduce significant bias and inaccuracies in synthetic samples, affecting the overall robustness and representativeness of the data. To tackle this issue effectively, synthetic data generation should be tailored to rebalance the class distribution while simultaneously reducing the influence of extreme data points.

This manuscript explores the fundamental concept of generating synthetic data with minimal impact from abnormal instances, recognizing its significance in various data-driven applications. By mitigating the influence of abnormal instances, we can improve the quality of synthetic data, ensuring that it is more reliable, dependable, and suitable for tasks such as machine learning, predictive modelling, and data augmentation.

By prioritizing the creation of robust synthetic data that is balanced and outlier-resistant, we can improve the performance of machine learning models in class imbalance problems. Upon visual inspection of our findings, we can see that the proposed tools significantly diminish the presence of data points drifting toward abnormal instances. Synthetic samples that accurately represent the minority class without being overly influenced by abnormal instances can lead to better generalization, more precise model predictions, and, ultimately, more reliable decision-making in applications like fraud detection, medical diagnosis, or rare-event forecasting. In this manuscript, we explore various strategies and techniques to achieve this dual objective, emphasizing the significance of generating synthetic data that minimizes the impact of abnormal instances, particularly in class imbalance, thereby contributing to developing more effective and robust data-driven solutions.

We have proposed several methods that offer differing approaches to the problem. In real-world data sets, we have seen differing results based on different approaches. In practice, it is critical for machine learning practitioners to examine different approaches to determine if any provide an advantage over the others.

Proposing multiple algorithms addresses the diverse characteristics of data, enabling tailored solutions for different scenarios. Our empirical evaluation highlights the strengths of each method, demonstrating that the optimal approach depends on the specific structure of the data. By offering various options and tools, we empower data users to accommodate varying complexities, dimensionalities, imbalance situations, nature of

outliers, and underlying structure. Then they can use multiple accuracy measures to determine which method performs best for particular needs as we have illustrated in the applications.

Given that results can vary significantly with different classifiers, selecting the appropriate classifier for the dataset is crucial. The choice of classifier can greatly impact the performance and accuracy of the model. For example, in our study, we used three classifiers: Logistic Regression, k-Nearest Neighbors (k-NN), and Random Forest. Notably, Dirichlet ExtSMOTE outperforms most other SMOTE variants in terms of F1 score, MCC, and PR-AUC, especially with a Random Forest classifier.

Random Forest is often more effective for imbalanced data due to its ensemble approach, combining multiple decision trees to enhance accuracy and robustness. This helps capture complex patterns and interactions within the data, which is crucial for imbalanced datasets. Additionally, Random Forest is more resilient to outliers and noise compared to Logistic Regression, which assumes linear relationships, and k-NN, which can struggle with high-dimensional data and noise. Studies show that Random Forest generally achieves higher accuracy and better generalization in imbalanced datasets compared to Logistic Regression and k-NN (Couronné et al.; 2018; Shah et al.; 2020). However, this highlights the need to experiment with multiple classifiers and select the one that best captures the underlying patterns in the data, ensuring robust and reliable outcomes.

Use of synthetic data has possible implications for safety and bias. This is especially relevant when working with human data, including medical and social data. There is evidence of bias that can be introduced into systems for the related problem of data augmentation (Jain et al.; 2018). Additionally, it has been observed that tools such as up- and down-sampling and augmentation are potential tools to combat bias in datasets, though this remains challenging (Tomalin et al.; 2021; Pastaltzidis et al.; 2022; Sharma et al.; 2020). This is an area of ongoing research, but practitioners should exert caution when employing tools for data imbalance or data augmentation, both with the output variables and with informative features.

5.6.1 Practical Implications

The enhancements to SMOTE for handling imbalanced data with abnormal minority instances have significant practical implications across various domains, including healthcare, finance, and fraud detection. By introducing advanced SMOTE techniques that are specifically designed to address the presence of abnormal instances within minority

classes, our study provides robust tools for improving the accuracy and reliability of predictive models. These enhanced methods facilitate better detection of rare but critical events, such as identifying rare diseases, detecting fraudulent transactions, or uncovering defective products. As a result, organizations can make more informed decisions, reduce risks, and optimize resource allocation. Implementing these methods empowers data practitioners to effectively tackle complex imbalanced data scenarios, ultimately leading to more precise and impactful outcomes.

5.6.2 Theoretical Foundations of Proposed Methods

5.6.2.1 Distance ExtSMOTE

Using normalized inverse distances as weights in Distance ExtSMOTE provides a straightforward and effective mechanism for generating synthetic samples that maintain the integrity of the minority class distribution. By assigning higher weights to closer neighbours, the method captures local density variations and preserves the underlying structure of the minority class. This approach also mitigates the impact of outliers, as distant outliers receive smaller weights, reducing their influence on synthetic samples. Consequently, synthetic instances are more representative of the minority class, enhancing the classifier's ability to generalize. Normalizing the inverse distances ensures that the sum of the weights is 1, maintaining balanced contributions from relevant neighbours and minimizing noise. This leads to lower bias in the synthetic samples and improved model performance on imbalanced datasets.

This method is suitable for datasets where the minority class has a relatively dense and well-defined structure. However, this approach has limitations. The deterministic nature of using fixed normalized inverse distances can result in less diverse synthetic samples, potentially missing the complexity of the minority class distribution.

5.6.2.2 Dirichlet ExtSMOTE

Using Dirichlet samples to define weights in Dirichlet ExtSMOTE, rather than relying solely on inverse distances, offers several significant advantages. The primary benefit is the introduction of variability in the synthetic sample generation process. By drawing weights from a Dirichlet distribution parameterized uniformly or by inverse distances, the method ensures that the weights can differ each time, producing a more diverse set of synthetic samples even for the same set of neighbours. This variability helps better approximate the true underlying distribution of the minority class, thereby significantly enhancing the generalization capability of the classifier. The Dirichlet distribution also

ensures that the weights sum to one and are non-negative, promoting a balanced and realistic generation of synthetic instances. This method helps distribute synthetic samples more evenly across the feature space, avoiding the clustering problem seen in SMOTE.

Additionally, the Dirichlet distribution provides a probabilistic framework that naturally emphasizes closer neighbours while maintaining controlled randomness. This reduces the risk of overfitting to specific patterns in the minority class and mitigates the impact of outliers, which are assigned lower weights due to their larger distances. This also helps balance the bias-variance tradeoff, leading to synthetic instances that are more representative of the minority class distribution and ultimately improving the performance of classifiers on imbalanced datasets.

This method is ideal for datasets where diversity among synthetic samples is crucial and the minority class has a complex, variable structure. It is particularly useful when overfitting needs to be mitigated. However, the Dirichlet ExtSMOTE approach may struggle with extremely sparse datasets where the minority class instances are very few and far apart. In such cases, the variability in the weights might not be sufficient to generate meaningful synthetic samples, and the overall benefit of the method may be reduced.

5.6.2.3 FCRP SMOTE

Using the finite Chinese Restaurant Process (FCRP) in FCRP SMOTE offers a sophisticated approach to generating synthetic samples by introducing adaptive weighting through iterative probability adjustments. Initially, neighbours are selected based on inverse distance probabilities, which are dynamically adjusted with an alpha value over multiple iterations. This method respects the natural clustering of the minority class, enhances the diversity of synthetic samples, and better captures the true distribution and local density variations of the minority class while mitigating the influence of outliers. By iteratively refining the weights, the method balances the bias-variance tradeoff, leading to more representative synthetic instances and improved classifier performance on imbalanced datasets.

This method is well-suited for datasets where the minority class has varying densities and complex local structures. Its iterative adjustment mechanism is beneficial for applications requiring adaptive sampling that reflects the nuanced distribution of the minority class. However, this approach also has limitations. The iterative adjustment process can be computationally intensive, especially for large datasets, due to the repeated recalculations of probabilities. Additionally, in extremely sparse datasets where minority class instances are few and far apart, the clustering mechanism might struggle to form

meaningful clusters, reducing the effectiveness of synthetic sample generation. Despite these challenges, FCRP SMOTE's adaptive mechanism provides significant advantages in generating high-quality synthetic samples.

5.6.2.4 BGMM SMOTE

Using BGMM SMOTE is advantageous because it leverages the probabilistic clustering capabilities of BGMMs to generate synthetic samples that closely reflect the true distribution of the minority class. By using mixing weights from the cluster where the minority median centroid belongs, the method ensures that synthetic instances are created in high-density regions of the minority class. This preserves the underlying structure and relationships among data points, leading to more accurate and diverse synthetic samples. This probabilistic framework also mitigates the impact of outliers, enhancing the classifier's ability to generalize from these synthetic instances.

This method is particularly suitable for datasets where the minority class can be effectively modeled using Gaussian mixtures, and where capturing the probabilistic distribution of the data is crucial. It is ideal for applications requiring detailed and accurate representation of the minority class distribution. However, the computational complexity of fitting a BGMM can be significant, particularly for large and high-dimensional datasets, making the process time-consuming. Additionally, in cases where minority class instances are sparse or do not form clear clusters, the model may struggle to capture the data distribution accurately.

In high-dimensional spaces, distance calculations can be computationally expensive and may not capture complex relationships. Using feature selection techniques beforehand can help mitigate the high-dimensionality issue.

The computations for this paper were performed using the resources provided by the Digital Research Alliance of Canada (Digital Research Alliance of Canada; 2024).

5.6.3 Future Research Directions

While our study introduces several promising methods, there are numerous avenues for future research that could further enhance these techniques and broaden their applications. Here are some suggestions:

- Extension to High-Dimensional Data: While our methods have shown promising results in bivariate scenarios, real-world applications often involve high-dimensional data. Future research should focus on adapting and optimizing these SMOTE extensions for high-dimensional datasets.

- Integration with Ensemble Learning Techniques: Combining the proposed SMOTE extensions with ensemble learning techniques, such as bagging, boosting, and stacking, could further improve classification performance. Future work could explore the synergy between these advanced sampling methods and ensemble classifiers to develop more robust and accurate models for imbalanced datasets.

- Ethical Considerations and Bias Mitigation: As highlighted in our discussion, the ethical implications of using synthetic data in sensitive applications need careful consideration. Future research should investigate techniques to identify and mitigate biases in synthetic data generation, ensuring that the models built on such data are fair and unbiased. This includes developing metrics and frameworks to evaluate the ethical impact of synthetic data on decision-making processes.

5.7 Conclusion

In summary, this manuscript serves as a foundational resource for comprehensively exploring solutions to enhance the applicability of SMOTE in the presence of abnormal instances within imbalanced datasets. By addressing this pivotal challenge, we aim to facilitate more robust and dependable classification in contexts where minority class instances hold paramount importance.

Bibliography

Asniar, Maulidevi, N. U. and Surendro, K. (2022). SMOTE-LOF for noise identification in imbalanced data classification, *Journal of King Saud University. Computer and information sciences* **34**(6): 3413–3423.

Barua, S., Islam, M. M., Yao, X. and Murase, K. (2014). Mwmote–majority weighted minority oversampling technique for imbalanced data set learning, *IEEE transactions on knowledge and data engineering* **26**(2): 405–425.

Barua, S., Islam, Md, M. and Murase, K. (2013). Prowsyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning, *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, pp. 317–328.

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

Bej, S., Davtyan, N., Wolfien, M., Nassar, M. and Wolkenhauer, O. (2021). Loras: an oversampling approach for imbalanced datasets, *Machine learning* **110**(2): 279–301.

Bela, A, F., Amol, K. and Maya, R, G. (2010). Introduction to the dirichlet distribution and related processes, *Technical report*, University of Washington Department of Electrical Engineering.

Blagus, R. and Lusa, L. (2013). Smote for high-dimensional class-imbalanced data, *BMC bioinformatics* **14**(1): 106–106.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for dirichlet process mixtures, *Bayesian analysis* **1**(1).

Breiman, L. (2001). Random forests, *Machine learning* **45**(1): 5–32.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: identifying density-based local outliers, *International Conference on Management of Data: Proceedings of the 2000 ACM SIGMOD international conference on Management of data : Dallas, Texas, United States; 15-18 May 2000*, ACM, New York, NY, USA, pp. 93–104.

Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C. (2012). DBSMOTE: Density-based synthetic minority over-sampling technique, *Applied intelligence (Dordrecht, Netherlands)* **36**(3): 664–684.

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

Chawla, N. V., Lazarevic, A., Hall, L. O. and Bowyer, K. W. (2003). SMOTEBoost: Improving Prediction of the Minority Class in Boosting, *Knowledge Discovery in Databases: PKDD 2003*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 107–119.

Cieslak, D., Chawla, N. and Striegel, A. (2006). Combating imbalance in network intrusion datasets, *2006 IEEE International Conference on Granular Computing*, pp. 732–737.

Couronné, R., Probst, P. and Boulesteix, A.-L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment, *BMC bioinformatics* **19**(1): 270–270.

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

Digital Research Alliance of Canada (2024). Digital Research Alliance of Canada, `https://alliancecan.ca/`.

Duan, L., Xu, L., Guo, F., Lee, J. and Yan, B. (2007). A local-density based spatial clustering algorithm with noise, *Information systems (Oxford)* **32**(7): 978–986.

Duin, R. (2024). Multiple Features, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5HC70.

Dunn, O. J. (1964). Multiple comparisons using rank sums, *Technometrics* **6**(3): 241–252.

Feng, S., Keung, J., Zhang, P., Xiao, Y. and Zhang, M. (2022). The impact of the distance metric and measure on smote-based techniques in software defect prediction, *Information and software technology* **142**: 106742–.

Fernández, A., del Jesus, M. J. and Herrera, F. (2010). On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets, *Information sciences* **180**(8): 1268–1291.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *Journal of the American Statistical Association* **32**(200): 675–701.

Gao, M., Hong, X., Chen, S. and Harris, C. J. (2011). On combination of smote and particle swarm optimization based radial basis function classifier for imbalanced problems, *The 2011 International Joint Conference on Neural Networks*, IEEE, pp. 1146–1153.

German, B. (1987). Glass Identification, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5WW2P.

Haberman, S. (1999). Haberman's Survival, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XK51.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning, *Advances in Intelligent Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 878–887.

Hawkins, D. (1980). *Identification of Outliers*, Monographs on Applied Probability and Statistics, Springer Netherlands, Dordrecht.

He, H., Bai, Y., Garcia, E. and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning, *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Vol. 10, IEEE, pp. 1322–1328.

Holm, S. (1979). A simple sequentially rejective multiple test procedure, *Scandinavian journal of statistics* **6**(2): 65–70.

Hu, S., Liang, Y., Ma, L. and He, Y. (2009). Msmote: Improving classification performance when training data is imbalanced, *2009 Second International Workshop on Computer Science and Engineering*, Vol. 2, IEEE, pp. 13–17.

Jain, N., Manikonda, L., Hernandez, A. O., Sengupta, S. and Kambhampati, S. (2018). Imagining an engineer: On gan-based data augmentation perpetuating biases, *arXiv.org* .

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

Lee, Hansoo; Kim, J. and Kim, S. (2017). Gaussian-based smote algorithm for solving skewed class distributions, *Int. J. Fuzzy Log. Intell. Syst.* **17**: 229–234.

Lemaitre, G., Nogueira, F. and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, *Journal of machine learning research* **18**: 1–5.

Little, M. (2008). Parkinsons, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C59C74.

Lohweg, V. (2013). Banknote Authentication, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55P57.

Low Resolution Spectrometer (1988). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5B02R.

Mahmoudi, S., Moradi, P., Akhlaghian, F. and Moradi, R. (2014). Diversity and separable metrics in over-sampling technique for imbalanced data classification, *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 152–158.

Matharaarachchi, S., Domaratzki, M. and Muthukumarana, S. (2021). Assessing feature selection method performance with class imbalance data, *Machine learning with applications* **6**: 100170–.

Mathew, J., Luo, M., Pang, C. K. and Chan, H. L. (2015). Kernel-based smote for svm classification of imbalanced datasets, *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 001127–001132.

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A. and Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance, *Neural networks : the official journal of the International Neural Network Society* **21**(2-3): 427—436.

Nakai, K. (1996). Yeast, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5KG68.

Nash, W., Sellers, T., Talbot, S., Cawthorn, A. and Ford, W. (1995). Abalone, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55C7W.

Newman, C. B. D. and Merz, C. (1998). UCI repository of machine learning databases.

Ng, K. W., Tian, G.-L. and Tang, M.-L. (2011). *Dirichlet and related distributions : theory, methods and applications*, Wiley Series in Probability and Statistics, Wiley, Chichester, England.

Pastaltzidis, I., Dimitriou, N., Quezada-Tavarez, K., Aidinlis, S., Marquenie, T., Gurzawska, A. and Tzovaras, D. (2022). Data augmentation for fairness-aware machine learning: Preventing algorithmic bias in law enforcement systems, *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ACM, New York, NY, USA, pp. 2302–2314.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

Pitman, J. and Picard, J. (2006). *Combinatorial stochastic processes : Ecole d'ete de probabilities de Saint-Flour XXXII - 2002*, Lecture notes in mathematics ; 1875, 1st ed. 2006. edn, Springer, Berlin ;.

Roberts, S., Husmeier, D., Rezek, I. and Penny, W. (1998). Bayesian approaches to gaussian mixture modeling, *IEEE transactions on pattern analysis and machine intelligence* **20**(11): 1133–1142.

Shah, K., Patel, H., Sanghvi, D. and Shah, M. (2020). A comparative analysis of logistic regression, random forest and knn models for the text classification, *Augmented human research* **5**(1).

Sharma, S., Zhang, Y., Ríos Aliaga, J. M., Bouneffouf, D., Muthusamy, V. and Varshney, K. R. (2020). Data augmentation for discrimination prevention and bias disambiguation, *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, ACM, New York, NY, USA, pp. 358–364.

Srinivasan, A. (1993). Statlog (Landsat Satellite), UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55887.

Suh, Y., Kim, C., Song, L., Yu, J. and Mo, J. (2017). A comparison of oversampling methods on imbalanced topic classification of korean news articles, *Journal of Cognitive Science* **18**: 391–437.

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.

Tomalin, M., Byrne, B., Concannon, S., Saunders, D. and Ullmann, S. (2021). The practical ethics of bias reduction in machine translation: why domain adaptation is better than data debiasing, *Ethics and information technology* **23**(3): 419–433.

Turlapati, V. P. K. and Prusty, M. R. (2020). Outlier-smote: A refined oversampling technique for improved detection of covid-19, *Intelligence-based medicine* **3-4**: 100023–100023.

Weisberg, S. (2005). *Applied Linear Regression*, John Wiley and Sons, Incorporated, Hoboken.

Wolberg, W., Mangasarian, O., Street, N. and Street, W. (1995). Breast Cancer Wisconsin (Diagnostic), UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5DW2B.

Yang, Z., Tang, W., Shintemirov, A. and Wu, Q. (2009). Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **39**: 597 – 610.

Zhu, B., Baesens, B., Backiel, A. and vanden Broucke, S. K. L. M. (2018). Benchmarking sampling techniques for imbalance learning in churn prediction, *Journal of the Operational Research Society* **69**(1): 49–65.

Chapter 6

Deep-ExtSMOTE: Integrating Autoencoders for Advanced Mitigation of Class Imbalance in High-Dimensional Data Classification

The content of this chapter is the manuscript text for to be submitted publication under the following citation:

Matharaarachchi, S., Domaratzki, M. and Muthukumarana, S. (2024). Deep-ExtSMOTE: Integrating Autoencoders for Advanced Mitigation of Class Imbalance in High-Dimensional Data Classification (In Review).

Deep-ExtSMOTE: Integrating Autoencoders for Advanced Mitigation of Class Imbalance in High-Dimensional Data Classification

Surani Matharaarachchi[a,*], Mike Domaratzki[b], Saman Muthukumarana[a]

[a] *Department of Statistics, University of Manitoba, Winnipeg, MB, Canada*
[b] *Department of Computer Science, Western University, London, ON, Canada*

Abstract

Class imbalance and the curse of dimensionality present significant challenges in machine learning, particularly in high-dimensional classification tasks. This study introduces Deep-ExtSMOTE, a novel technique designed to address these challenges by integrating autoencoder-based dimensionality reduction with an Extended version of SMOTE. Deep-ExtSMOTE combines the power of autoencoders, which effectively reduce data dimensionality and capture complex, non-linear relationships, with the advanced resampling capabilities of Extended SMOTE, which generates synthetic samples through weighted averages of neighboring points. This integration aims to enhance the representation of minority classes and improve the overall robustness of classification models. Our empirical evaluation demonstrates that Deep-ExtSMOTE substantially outperforms traditional SMOTE and autoencoders alone, particularly in high-dimensional datasets. The method exhibits significant improvements in classification performance, as measured by F1 scores, and effectively addresses the limitations of existing techniques. Additionally, our results highlight that the improvement with Deep-ExtSMOTE compared to SMOTE and autoencoders individually is notable, emphasizing the method's potential to enhance model accuracy and reliability.

Keywords: *Class Imbalance, Curse of Dimensionality, Autoencoders, SMOTE*

---

*Corresponding Author

Email addresses: matharas@myumanitoba.ca (Surani Matharaarachchi), mdomarat@uwo.ca (Mike Domaratzki), saman.muthukumarana@umanitoba.ca (Saman Muthukumarana)

6.1 Introduction

In recent years, advancements in data collection techniques have led to an exponential increase in the availability of high-dimensional datasets across various domains, ranging from healthcare to finance. However, alongside this wealth of information comes the challenge of analyzing datasets where the number of features exceeds the number of samples, a scenario commonly referred to as the "large p, small n" problem (Huynh et al.; 2020). This imbalance fundamentally alters the dynamics of traditional statistical and machine learning methods, often leading to biased model learning and diminished predictive performance.

Moreover, high-dimensional datasets frequently encounter class imbalance, where one or more classes are underrepresented compared to others (Krawczyk; 2016). This imbalance skews the learning process of classifiers, leading to biased predictions and reduced performance in minority class instances. Traditional resampling techniques such as oversampling and undersampling attempt to mitigate this issue, but they often need to pay more attention to the inherent distributional characteristics of the dataset, leading to suboptimal performance.

Feature reduction emerges as a critical strategy to address the challenges of high dimensionality and class imbalance (Garzon; 2022). By selecting a subset of informative features or transforming the original feature space into a lower-dimensional representation, feature reduction techniques aim to enhance model interpretability, reduce computational burden, and improve prediction accuracy.

Despite the advancements in feature reduction and resampling techniques, abnormal instances within the minority class further complicate the learning process. These outliers or noisy samples distort decision boundaries, leading to false predictions and diminished model reliability. Therefore, effective outlier detection and handling mechanisms are imperative to ensure robust model performance.

The Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al.; 2002) is a promising class balancing approach, but it generates noise when synthetic samples are created. Suja and Nayahi (2023) propose a method that integrates feature extraction before data sampling, utilizing techniques like autoencoder to reduce noise generated by Synthetic Minority Oversampling Technique (SMOTE) for class balancing,

followed by classification using Deep one-dimensional Convolutional Neural Network. Salekshahrezaee et al. (2021) tackles the difficulty of training machine learning models on imbalanced and high-dimensional datasets by suggesting a method that incorporates feature extraction before data sampling. It compares principal component analysis (PCA) and convolutional autoencoder (CAE) for feature extraction and finds that CAE performs better. Combining CAE with SMOTE for data sampling achieves the highest F1-score of 90.5% when evaluated on a credit card fraud dataset using a random forest classifier.

In this paper, we present a novel rebalancing technique designed to address the complexities arising from high-dimensional data, class imbalance, feature reduction, and the presence of abnormal instances in minority classes. Our approach uses autoencoders for feature reduction, followed by ExtSMOTE (Matharaarachchi et al.; 2024) and related strategies to generate synthetic instances for the minority class. This combination aims to minimize the influence of abnormal instances on synthetic generation, intelligently synthesizing minority class samples while lowering the impact of sparse data.

To address a wide range of practical scenarios, we initially generate synthetic data using the "make_classification" function from the Python scikit-learn.datasets library (Pedregosa et al.; 2011). We then evaluate and compare the performance of both existing and proposed methods on these synthetic datasets. Subsequently, we apply these methods to a real-world dataset to draw additional insights. Classification was conducted using two different classifiers: Logistic Regression (Weisberg; 2005), and Random Forest (Breiman; 2001). To assess model performance, we employed several accuracy metrics, including F1-Score, Precision, Recall, PR-AUC, and MCC.

The paper is organized as follows. In this section, we discuss the challenges posed by class imbalance and high dimensionality in classification tasks, highlighting the limitations of existing techniques such as traditional SMOTE. We motivate the need for a novel approach that integrates dimensionality reduction with advanced resampling methods to improve model performance. Section 6.2 introduces Deep-ExtSMOTE, detailing the architecture that combines autoencoder-based feature reduction with an extended version of SMOTE. We explain how this integration effectively manages the curse of dimensionality and enhances synthetic sample generation for minority classes. Section 6.3 presents results on diverse high-dimensional and imbalanced simulated datasets. We demonstrate that Deep-ExtSMOTE significantly outperforms traditional methods in terms of classification accuracy and class balance. Finally, 6.4 discusses these findings, highlighting the strengths and potential limitations of Deep-ExtSMOTE.

6.2 Methodology

6.2.1 Autoencoders

An autoencoder (Goodfellow et al.; 2016) is a neural network architecture designed to learn compressed representations of input data by encoding it into a lower-dimensional space. This process is primarily used for dimensionality reduction and feature extraction, enabling the network to capture the most important patterns and features within the data while discarding redundant information. Autoencoders consist of two main parts: an encoder and a decoder. The encoder compresses the input data into a low-dimensional representation, while the decoder reconstructs the original data from this representation.

In the context of a multi-layer autoencoder, the encoder consists of several layers that sequentially reduce the dimensionality of the input $\mathbf{x}^{(i)}$, where i denotes the i^{th} observation. At each layer of the encoder, input features are multiplied by a weight matrix and passed through an activation function, introducing non-linearity. This process allows the model to learn complex, non-linear mappings. The transformation at each layer can be mathematically expressed as:

$$\begin{aligned}
\mathbf{h}_1^{(i)} &= f_1(\mathbf{W}_1^T \cdot \mathbf{x}^{(i)} + \mathbf{b}_1) \\
\mathbf{h}_2^{(i)} &= f_2(\mathbf{W}_2^T \cdot \mathbf{h}_1^{(i)} + \mathbf{b}_2) \\
&\dots \\
\mathbf{z}^{(i)} &= f_n(\mathbf{W}_n^T \cdot \mathbf{h}_{n-1}^{(i)} + \mathbf{b}_n)
\end{aligned}$$

where $\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, \dots$ are the intermediate representations at each layer. $\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_n$ are the weight matrices for each layer, where $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$ are the biased vectors for each layer. $\mathbf{z}^{(i)}$ is the latent vector, representing the lower-dimensional feature space and $f_1, f_2, \dots, f_n$ represent the non-linear activation functions used in each layer.

The decoder typically applies a process similar to the encoder but in reverse. It takes the latent vector $\mathbf{z}^{(i)}$ and maps it back to a reconstruction of the original input. The process at each layer of the decoder can be expressed as:

$$\begin{aligned}
\mathbf{h'}_{n-1}^{(i)} &= f'_{n-1}(\mathbf{W'}_{n-1}^T \cdot \mathbf{z}^{(i)} + \mathbf{b'}_{n-1}) \\
\mathbf{h'}_{n-2}^{(i)} &= f'_{n-2}(\mathbf{W'}_{n-2}^T \cdot \mathbf{h'}_{n-1}^{(i)} + \mathbf{b'}_{n-2}) \\
&\dots \\
\hat{\mathbf{x}}^{(i)} &= f'_1(\mathbf{W'}_1^T \cdot \mathbf{h'}_2^{(i)} + \mathbf{b'}_1)
\end{aligned}$$

where $\hat{\mathbf{x}}^{(i)}$ is the constructed output for the i^{th} observation, which approximates $\mathbf{x}^{(i)}$. $\mathbf{h'}^{(i)}_{n-1}, \mathbf{h'}^{(i)}_{n-2}, \ldots$ are the intermediate representations in the decoder. $\mathbf{W'}_1, \mathbf{W'}_2, \ldots, \mathbf{W'}_{n-1}$ are the weight matrices for each decoder layer, where $\mathbf{b'}_1, \mathbf{b'}_2, \ldots, \mathbf{b'}_{n-1}$ are the biased vectors for each decoder layer. $\mathbf{z'}^{(i)}$ is the latent vector, representing the lower-dimensional feature space and $f'_1, f'_2, \ldots, f'_{n-1}$ represent the non-linear activation functions used in each decoder layer.

To train the multi-layer autoencoder, we use a loss function that measures the difference between the input $\mathbf{x}^{(i)}$ and the reconstructed output $\hat{\mathbf{x}}^{(i)}$. The most common loss function for this purpose is Mean Squared Error (MSE):

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}||^2 \tag{6.1}$$

where n is the total number of observations in the dataset. By minimizing this loss function, the autoencoder learns to retain the most important features in the latent representation $\mathbf{z}^{(i)}$ while accurately reconstructing the input data.

Autoencoders are well-suited for datasets that have high dimensionality, co-linearity, complex non-linear relationships, and unknown structures. However, the performance of autoencoders is significantly influenced by the choice of activation functions used in both the encoder and decoder. Activation functions introduce non-linearity into the model, allowing it to capture complex patterns and relationships within the data. By incorporating non-linear activation functions like sigmoid or ReLU, autoencoders can effectively reduce the dimensionality of the data while capturing intricate patterns and structures. This allows them to achieve a more robust generalization than traditional methods such as PCA (Zhang and Bom; 2021).

Common activation functions in the encoder include Rectified Linear Units (ReLU), Sigmoid, and Tanh. ReLU is popular for its simplicity and effectiveness in handling non-linearity, providing a piecewise linear function that helps in learning sparse representations. Sigmoid and Tanh functions, on the other hand, are useful for compressing values into a bounded range, which can be advantageous when the data is normalized or when smoother activation is needed. The choice of activation function in the encoder affects the quality of the compressed representation and its ability to capture critical features of the data. In the decoder, the choice of activation function influences how well the reconstructed data matches the original input.

By selecting appropriate activation functions, autoencoders can be optimized to better capture and reconstruct the underlying structure of the data, thereby enhancing the

overall effectiveness of the dimensionality reduction and feature learning processes. This choice directly impacts the quality of the low-dimensional representation used by techniques like Deep-ExtSMOTE for generating synthetic samples and addressing the class imbalance.

6.2.2 SMOTE with Autoencoders

Integrating SMOTE with the trained encoder introduces an approach to resampling techniques, capitalizing on the intrinsic representations learned by the autoencoder. By leveraging the encoded representations, which encapsulate the essential features of the input data, SMOTE can generate synthetic samples that are not only balanced in class distribution but also semantically meaningful. Unlike traditional resampling techniques that operate directly on the input space, SMOTE applied to encoded representations benefits from the compact and informative representations extracted by the encoder. This relationship between autoencoder-based feature learning and SMOTE enables the creation of synthetic samples more representative of the underlying data manifold, thereby enhancing the diversity and quality of the resampled dataset.

Furthermore, using SMOTE with the trained encoder helps mitigate the risk of overfitting inherent in traditional resampling methods. By operating in the latent space learned by the autoencoder, SMOTE ensures that the generated synthetic samples adhere to the underlying data distribution captured by the encoder. This not only aids in keeping the intrinsic structure of the data but also reduces the likelihood of introducing spurious patterns or artifacts during the resampling process. Consequently, the resampled dataset obtained through this approach is more robust and conducive to training classifiers that generalize well to unseen data.

Moreover, incorporating SMOTE with the trained encoder facilitates the seamless integration of feature learning and data augmentation within a unified framework. By jointly optimizing the encoder and SMOTE parameters, practitioners can tailor the resampling process to the specific characteristics of the dataset and the requirements of the downstream classification task. This approach streamlines the model development pipeline and fosters synergistic interactions between different components, enhancing model performance and interpretability.

6.2.3 Deep-ExtSMOTE Architecture: Dirichlet ExtSMOTE with Autoencoders

Deep-ExtSMOTE is a novel technique designed to improve classification performance in high-dimensional datasets, particularly when dealing with underrepresented classes. The approach integrates autoencoder-based feature reduction with an Extended version of the Synthetic Minority Over-sampling Technique (SMOTE). Fig. 6.1, Fig. 6.2, and Fig. 6.3 illustrate the architecture of our approach in three main steps: 1). Training the Autoencoder, 2). Extracting the Encoder Representation, and 3) Resampling and Classification.

This section provides a detailed overview of each step.

1. **Train the Autoencoder**

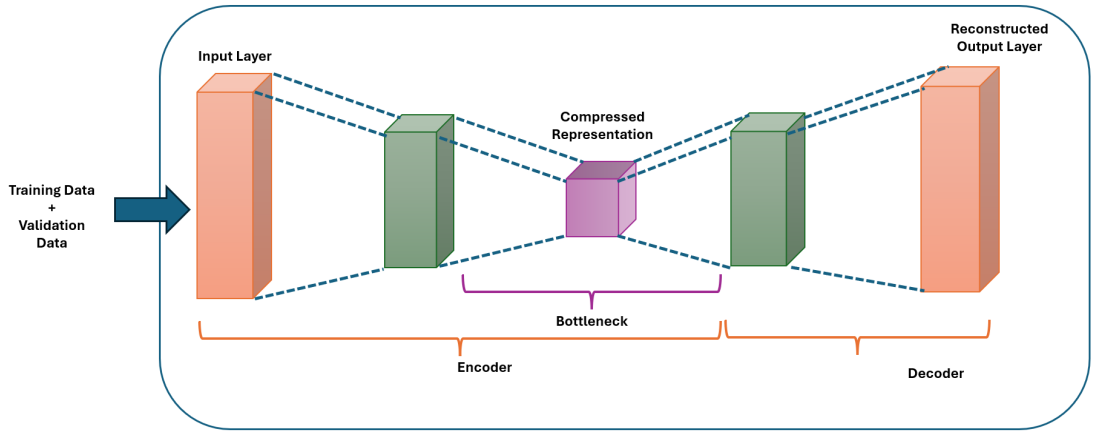


Figure 6.1: Train the Autoencoder

The first step involves training an autoencoder to reduce the dimensionality of the high-dimensional dataset. Autoencoders are neural network models designed to learn a compressed, lower-dimensional representation of the input data. This process involves two main components: the encoder and the decoder.

Encoder: The encoder part of the autoencoder takes the input data and maps it to a lower-dimensional representation. In this case, the encoder is implemented using a Sequential model with three dense layers:

- The first layer (Dense) has 128 units and uses the ReLU (Rectified Linear Unit) activation function, which introduces non-linearity to the network.
- The second layer has 64 units and also uses the ReLU activation function.

- The third layer also has 32 units and uses the ReLU activation function.

The input shape of the encoder is defined by the 'input_dim' parameter, which is the total number of features in the original dataset.

Decoder: The decoder part of the autoencoder takes the low-dimensional representation produced by the encoder and reconstructs the original input data. In this case, the decoder is implemented using another Sequential model with three dense layers:

- The first layer has 64 units and uses the ReLU activation function. It takes the low-dimensional representation produced by the encoder, which has 32 units, as input.
- The second layer has 128 units and uses the ReLU activation function.
- The third layer has 'input_dim' units (same as the original input dimensionality) and uses the sigmoid activation function. Sigmoid is used for the final layer when the output needs to be normalized between 0 and 1.

Training the autoencoder involves minimizing the reconstruction error, that is the difference between the original and reconstructed data. This is achieved using the optimization technique called stochastic gradient descent. The goal is to ensure that the encoder produces meaningful, compact representations of the data that preserve its critical features.

After setting up the encoder and decoder components, they are combined to create the autoencoder model by placing them in a list within the Sequential constructor. Next, the autoencoder model uses the compile method to define the optimizer and loss function for training. In this case, we utilize the 'adam' optimizer, known for its efficiency and effectiveness across different problems. We opt for Mean Squared Error as the loss function, a common choice for regression problems like autoencoder reconstruction.

Once the model is compiled, it undergoes training using the provided training data. We specify parameters such as the number of epochs (set to 50) and batch size (set to 32).

2. **Extracting the Encoder Representation**

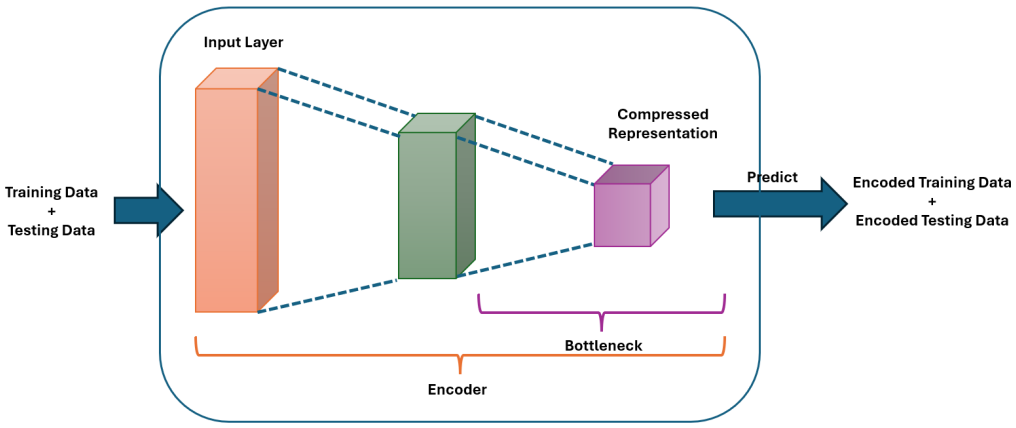


Figure 6.2: Extract encoded representation

Upon training completion, we use the encoder part of the model to extract the encoded representations of the input data. This involves using the trained encoder to transform the original high-dimensional data into its lower-dimensional representation.

- **Feature Extraction:**

  The encoder's output provides a compressed version of the input data, capturing the essential features needed for effective classification. This reduced representation simplifies the data, making it more manageable for subsequent processing and analysis.

- **Dimensionality Reduction:**

  By operating in this lower-dimensional space, Deep-ExtSMOTE addresses the curse of dimensionality, which often complicates learning algorithms and can lead to overfitting. The reduced representation highlights the intrinsic patterns and relationships within the data, facilitating better handling of class imbalance and abnormal instances.

3. **Resampling and Classification**

Figure 6.3: Resampling and Classification

The final step involves applying the Extended SMOTE algorithm to the encoded representation to generate synthetic samples. Subsequently, we employ the encoded representations as input features for a logistic regression classifier. The classifier is trained using the encoded training data alongside their corresponding labels. After training, the trained classifier makes predictions on the test data that is first passed through the encoder, thus completing the classification task.

- **Resampling:** Deep-ExtSMOTE uses the Extended version of SMOTE (Matharaarachchi et al.; 2024) to create synthetic samples for the minority class. This involves generating new data points by taking weighted averages of neighboring samples in the lower-dimensional space. The weighting is based on the distance between samples and aims to produce synthetic instances representative of the minority class's distribution.
- **Classification:** The synthetic samples, along with the original data, are then used to train a classification model. This model benefits from the improved class balance and the enhanced representation of minority classes provided by Deep-ExtSMOTE. The classification process involves applying standard machine learning algorithms to the resampled dataset to make predictions and assess model performance.

In summary, the three steps of Deep-ExtSMOTE—Training the Autoencoder, Extracting the Encoder Representation, and Resampling and Classification—work together to address the challenges of class imbalance and high-dimensional data. By leveraging autoencoders for dimensionality reduction and employing an advanced SMOTE variant for synthetic sample generation, Deep-ExtSMOTE enhances the accuracy and effectiveness of classification models in complex, real-world applications.

6.2.3.1 Components of the Deep Learning Framework

In implementing Deep-ExtSMOTE, TensorFlow's Keras library (Abadi et al.; 2016; Chollet; 2018) is used to construct and train deep learning models, particularly autoencoders, critical in generating synthetic samples within the latent space. The layers module from Keras is employed to define the architecture of the autoencoder, including the input layer, hidden layers, and the output layer, thereby enabling the compression and reconstruction of high-dimensional data. The models module compiles and manages the overall autoencoder model, facilitating the training process.

Additionally, to prevent overfitting during training, the EarlyStopping callback is integrated. This callback monitors the validation loss and halts training when the loss no longer improves, restoring the model to the state with the best performance. By incorporating these tools, Deep-ExtSMOTE effectively leverages deep learning techniques to enhance the synthesis of minority class samples, ensuring more robust and accurate classification in imbalanced datasets.

6.2.4 Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique used to transform a high-dimensional dataset into a lower-dimensional form while retaining as much variance as possible. The goal is to identify the most important features (principal components) that capture the most significant variation in the data.

Determining the optimal number of principal components in PCA is a crucial step to ensure that the dimensionality of the dataset is reduced effectively while retaining the essential information. In this study, we determined the optimal number of principal components using an automated method that relies on a predefined variance threshold. After calculating the cumulative explained variance for each principal component, we set a threshold at 90%, meaning we wanted to retain enough components to explain at least 90% of the total variance.

To find the exact number of components that meet this threshold in the simulation setup, we employed an approach where we identified the first principal component for which the cumulative explained variance reaches or exceeds the 90% mark. This was done by locating the position in the cumulative variance sequence where this condition is satisfied. Mathematically, this can be expressed as finding the smallest integer n such that:

$$\text{Cumulative Variance(n)} \geq \text{Threshold}$$

where the threshold is 0.90 (or 90%).

This automated method allowed us to precisely and efficiently determine the optimal number of components needed to retain the desired amount of variance, ensuring that our dimensionality reduction was both effective and aligned with our analytical goals.

6.3 Simulation Results

To identify the most effective method, the results were generated using various synthetic datasets with a sample size of 1000, featuring either 500 or 5000 features. For the 500-feature datasets, all features were informative, and we considered imbalance ratios (IR) of 3. For the 5000-feature datasets, 2000 features were informative, and the IR was set at 3. The evaluation included a baseline scenario without resampling and an original scenario without feature reduction. We applied two feature reduction techniques: autoencoders and PCA and fitted two machine learning classification methods: Logistic Regression (Logistic) and Random Forest Classifier (RFC) on the training data. Each dataset was split into two subsets: 80% for training and 20% for testing. Additionally, the training set was further divided into 80% for training and 20% for validation.

The output F1 Scores for two sample datasets are displayed in Fig. 6.4 and 6.5. In these figures, the first box plot to the left of the red dashed line represents the results of Dirichlet ExtSMOTE. The proposed Deep-ExtSMOTE (combining autoencoders with Dirichlet ExtSMOTE) is shown in the first column for two different classifiers. According to the figures, autoencoders enhance the performance of Dirichlet ExtSMOTE, particularly with a higher number of features, including noisy and non-informative ones.

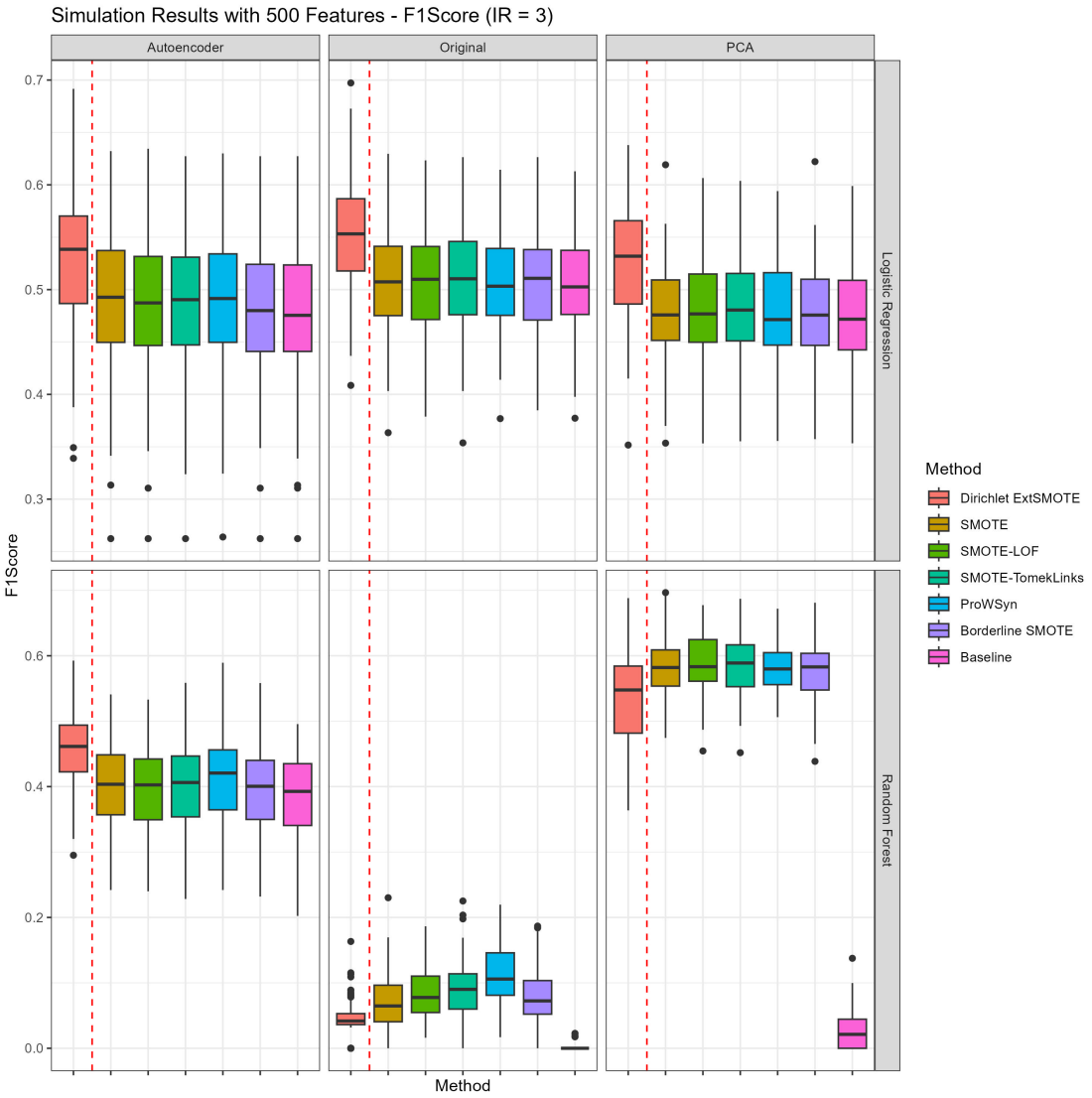


Figure 6.4: F1-Score distribution for 100 trials using simulated datasets with 1000 samples and 500 features (all informative), with an imbalance ratio (IR) of 3.

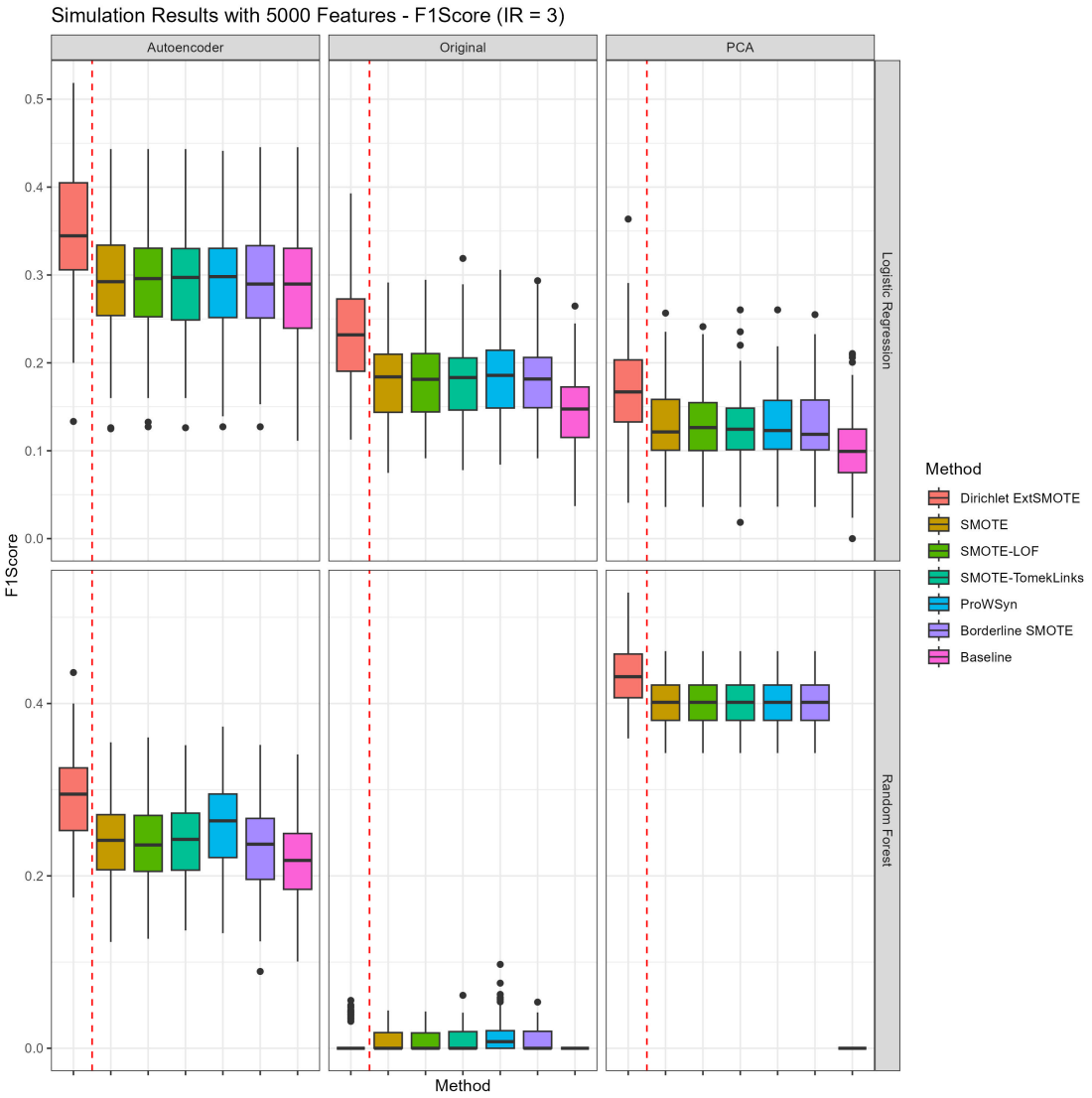


Figure 6.5: F1-Score distribution for 100 trials using simulated datasets with 1000 samples and 5000 features (2000 informative), with an imbalance ratio (IR) of 3.

6.4 Application Results

In this section, we present the results of applying the proposed deepExtSMOTE method to the Isolet dataset from the UCI Machine Learning Repository (Cole and Fanty; 1994; Pedregosa et al.; 2011). The dataset comprises recordings from 150 subjects, each of whom spoke the name of every letter in the alphabet twice, resulting in 52 training examples per speaker. For the purposes of this study, letters A and B were categorized as the minority class. The dataset includes 617 continuous features.

We conducted the analysis in two scenarios: a low-to-high dimensionality scenario, using all samples from the original dataset, and a moderate-to-high dimensionality scenario, where we reduced the sample size. The results for these two scenarios are discussed in Sections 6.4.1 and 6.4.2, respectively. Similar to the synthetic setup, we applied two feature reduction techniques: autoencoders and PCA. For all trials, the number of principal components (PCs) was set to 30, based on the scree plot analysis for both the original and reduced data.

6.4.1 Scenario 1: Original Isolet Dataset

The original dataset includes 7797 samples, resulting in a feature-to-sample ratio of approximately 0.0791 and an imbalance ratio of 13%. We employed a similar experimental setup as in the simulation study, running the experiment across 50 different training and test splits.

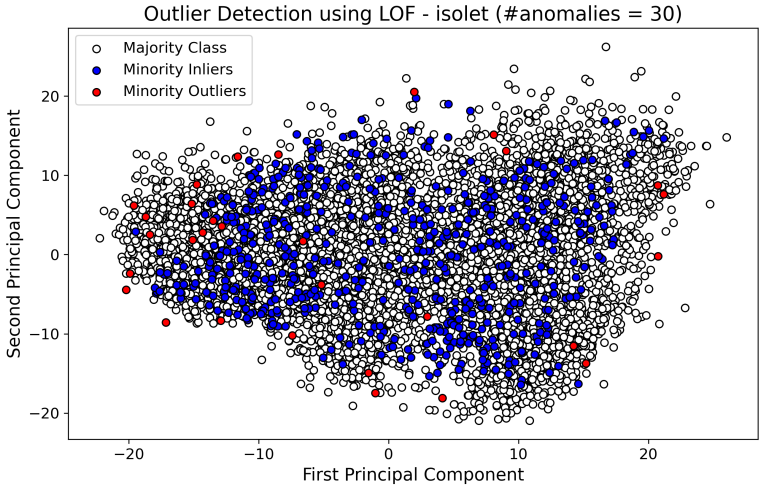


Figure 6.6: Scatter plot of the first two principal components in the original Isolet dataset, with the minority class in blue and detected outliers in red.

Fig. 6.6 shows a scatter plot of the first two principal components in the original Isolet dataset. The minority class is highlighted in blue, and minority outliers are marked in red. Local Outlier Factor (LOF) (Breunig et al.; 2000; Duan et al.; 2007) was used to detect these anomalies, with 30 such outliers identified in this example.

Fig. 6.7 illustrates the distribution of F1 Scores for two classifiers with autoencoders. These figures highlight performance consistency and variability among different machine learning models on the Isolet dataset.

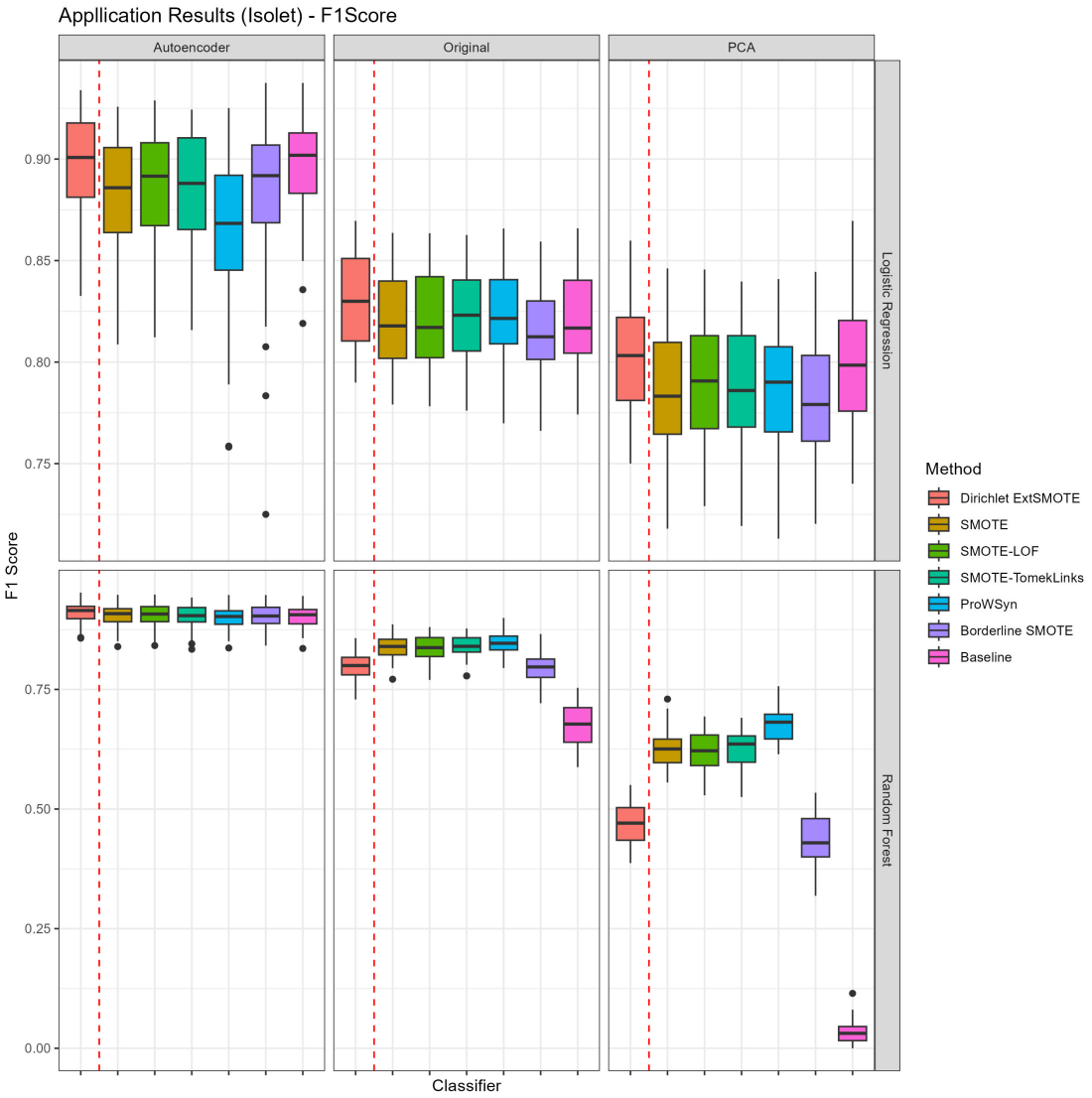


Figure 6.7: F1 Scores for the Isolet dataset across 50 training and test splits.

Table 6.1 compares the results of six different resampling techniques against the baseline, across two feature reduction techniques, original data, and two classifiers. Column-wise comparisons identify the best resampling technique, which is highlighted in bold. Row-wise comparisons assess the effectiveness of feature reduction techniques, with the best method for each performance measure highlighted in blue. The Deep-ExtSMOTE contribution, which combines Dirichlet ExtSMOTE and autoencoders, is indicated by row/column names highlighted in red. The last column shows the statistical significance of the mean difference of each performance measure between Dirichlet ExtSMOTE and the baseline, determined using the Wilcoxon signed-rank test at a 5% confidence level.

Table 6.1: Comparison of resampling and feature reduction methods for the original Isolet data. Best techniques and methods are highlighted, with Deep-ExtSMOTE in red and statistical significance in the last column.

Performance Measure	Feature Reduction Method	Classifier	Dirichlet ExtSMOTE	SMOTE	SMOTE-LOF	SMOTE-TomekLinks	ProWSyn	Borderline SMOTE	Baseline	Mean Difference Significance: Dirichlet ExtSMOTE vs. Baseline
F1 Score										
	Autoencoder	Logistic	**0.8971**	0.8822	0.8868	0.8832	0.8629	0.8799	0.896	No
	Autoencoder	RFC	**0.9088**	0.9036	0.904	0.9025	0.8984	0.9016	0.902	Yes
	Original	Logistic	**0.8302**	0.8199	0.8207	0.8215	0.822	0.8159	0.8196	Yes
	Original	RFC	0.7986	0.8388	0.8335	0.839	**0.8462**	0.7954	0.6758	Yes
	PCA	Logistic	**0.8022**	0.7853	0.7885	0.7872	0.787	0.7811	0.7983	Yes
	PCA	RFC	0.4731	0.626	0.6235	0.6262	**0.6753**	0.4335	0.0321	Yes
MCC										
	Autoencoder	Logistic	**0.8891**	0.8734	0.8781	0.8744	0.8545	0.8716	0.8881	No
	Autoencoder	RFC	**0.9015**	0.8958	0.8962	0.8946	0.8902	0.8938	0.8944	Yes
	Original	Logistic	**0.8164**	0.8054	0.8061	0.807	**0.8077**	0.801	0.8054	Yes
	Original	RFC	**0.789**	0.8264	0.8209	0.8268	**0.8341**	0.7846	0.6943	Yes
	PCA	Logistic	**0.7865**	0.7686	0.772	0.7705	**0.7703**	0.7635	0.7823	Yes
	PCA	RFC	0.5269	0.6351	0.6337	0.6354	**0.6748**	0.4794	0.1101	Yes
PR-AUC										
	Autoencoder	Logistic	**0.9014**	0.8876	0.8917	0.8885	0.8723	0.8871	0.9012	No
	Autoencoder	RFC	**0.9127**	0.9074	0.9078	0.9063	**0.9023**	0.906	0.9069	Yes
	Original	Logistic	**0.8369**	0.8272	0.8277	0.8286	0.8293	0.8232	0.8276	Yes
	Original	RFC	0.8183	0.8467	0.8418	0.8472	**0.853**	0.8134	0.7629	Yes
	PCA	Logistic	**0.8106**	0.7952	0.798	0.7968	0.7966	0.7903	0.8071	Yes
	PCA	RFC	0.6526	0.7031	0.703	0.7035	**0.7291**	0.6023	**0.5458**	Yes
Precision										
	Autoencoder	Logistic	0.8798	0.8554	0.866	0.8561	0.8119	0.8706	**0.9086**	Yes
	Autoencoder	RFC	0.9117	0.897	0.9	0.896	0.8812	0.9059	**0.9161**	Yes
	Original	Logistic	**0.8181**	0.8138	0.812	0.811	0.8155	0.7963	**0.8279**	Yes
	Original	RFC	0.8879	0.8579	0.8546	0.8613	0.8552	0.8777	**0.9685**	Yes
	PCA	Logistic	0.783	0.7495	0.7537	0.7506	**0.75**	0.7562	**0.798**	Yes
	PCA	RFC	**0.9352**	0.88	0.8824	0.8805	**0.8708**	0.8591	0.8571	Yes
Recall										
	Autoencoder	Logistic	0.9165	0.9133	0.9107	0.9144	**0.9271**	0.8957	0.8851	Yes
	Autoencoder	RFC	0.9066	0.9109	0.9086	0.9098	**0.9169**	0.8983	0.8892	Yes
	Original	Logistic	**0.8438**	0.8276	0.8305	0.8335	0.8302	0.8378	0.813	Yes
	Original	RFC	0.7278	0.8219	0.815	0.8193	**0.8385**	0.7284	0.5208	Yes
	PCA	Logistic	0.8249	0.8276	0.8294	0.83	**0.8302**	0.8099	0.8009	Yes
	PCA	RFC	0.3178	0.4869	0.4841	0.4872	**0.5533**	0.2914	0.0165	Yes

The analysis reveals that applying SMOTE extensions to the Isolet dataset results in a marginal improvement in the F1 score when combined with autoencoders (0.8971 with Dirichlet ExtSMOTE vs. 0.896 Baseline) for the logistic regression classifier with autoencoders. The statistical test confirmed that this difference in F1 score is not statistically significant, indicating that the slight improvement may be due to random variation rather than a meaningful enhancement in model performance. However, with the random forest classifier, the difference in F1 scores is statistically significant, indicating that the use of Deep-ExtSMOTE has a more substantial impact on model performance with this classifier.

Also, the recall values showed a significant difference. With Deep-ExtSMOTE (Dirichlet), the model achieved higher recall (0.9165) compared to the Baseline autoencoder (0.8851), and this difference was statistically significant. This suggests that Deep-ExtSMOTE meaningfully improves the model's ability to correctly identify positive instances in the minority class, making it more sensitive to detecting the minority class.

6.4.2 Scenario 2: Reduced Isolet Dataset

To introduce high-dimensional characteristics to the problem, we selected a subset of 1000 samples from the original 7797 samples. This adjustment resulted in a feature-to-sample ratio of 0.617, along with an imbalance ratio of 13%. We used a similar experimental setup as in the simulation study, running the experiment across 100 different training and test splits. Fig. 6.8 shows a scatter plot of the first two principal components in the reduced Isolet dataset.

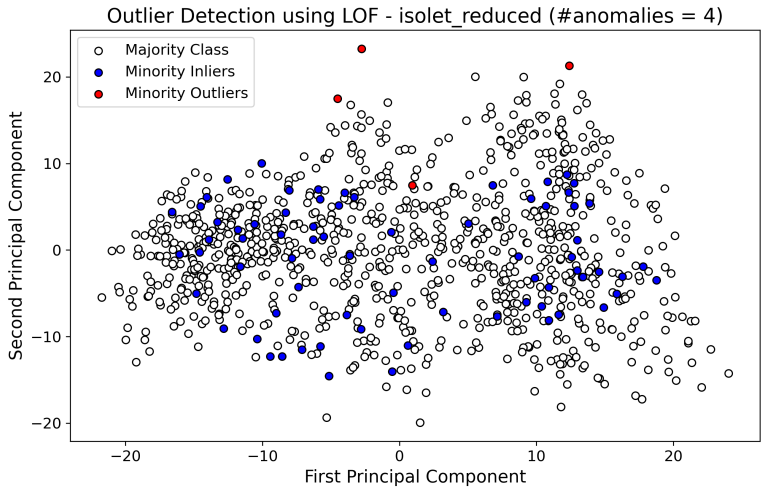


Figure 6.8: Scatter plot of the first two principal components in the reduced Isolet dataset, with the minority class in blue and detected outliers in red.

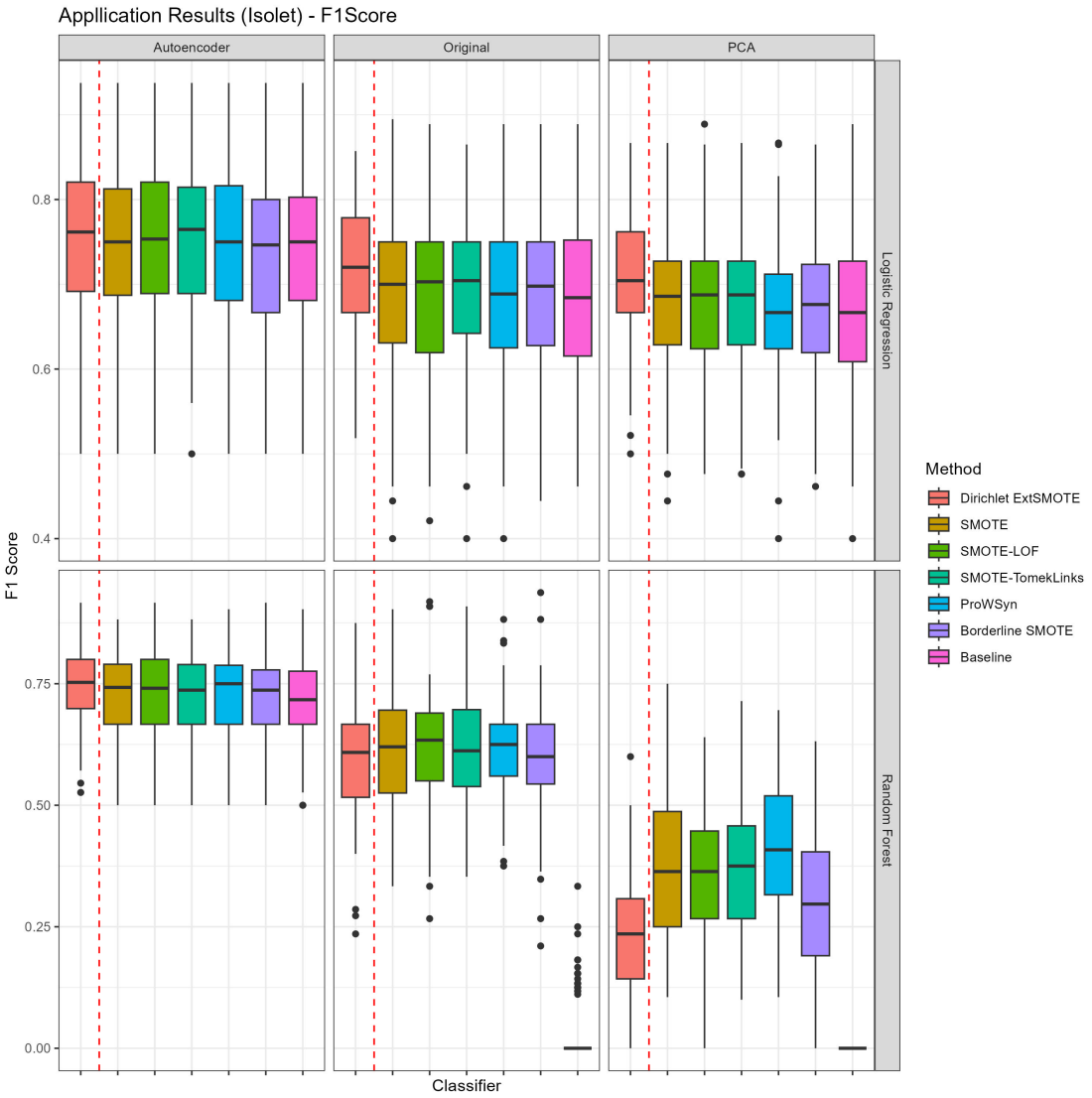


Figure 6.9: F1 Scores for the reduced Isolet dataset across 100 training and test splits.

Fig. 6.9 also illustrates the distribution of F1 scores for two classifiers applied to the reduced Isolet dataset, comparing performance across three scenarios: autoencoder-based feature reduction, PCA-based feature reduction, and without feature reduction. Table 6.2 presents results in a similar format to those in Table 6.1, facilitating a direct comparison of Section 6.4.1.

Table 6.2: Comparison of resampling and feature reduction methods for the reduced Isolet data. Best techniques and methods are highlighted, with Deep-ExtSMOTE in red and statistical significance in the last column.

Performance Measure	Feature Reduction Method	Classifier	Dirichlet ExtSMOTE	SMOTE	SMOTE-LOF	SMOTE-TomekLinks	ProWSyn	Borderline SMOTE	Baseline	Mean Difference Significance: Dirichlet ExtSMOTE vs. Baseline
F1 Score										
	Autoencoder	Logistic	**0.7528**	0.7469	0.7506	0.7495	0.7482	0.7443	0.7457	Yes
	Autoencoder	RFC	**0.7468**	0.7318	0.7318	0.7314	0.7337	0.7291	0.7189	Yes
	Original	Logistic	**0.7138**	0.6894	0.6859	0.6897	0.6815	0.6863	0.6771	Yes
	Original	RFC	0.5912	0.612	0.6123	0.6148	**0.6159**	0.5957	0.0307	Yes
	PCA	Logistic	**0.7041**	0.675	0.6771	0.6773	0.6702	0.6696	0.6658	Yes
	PCA	RFC	0.2359	0.3735	0.3559	0.3672	**0.4115**	0.3027	0	Yes
MCC										
	Autoencoder	Logistic	**0.7398**	0.733	0.7369	0.7354	0.7344	0.7313	0.7329	Yes
	Autoencoder	RFC	**0.7386**	0.722	0.7219	0.7204	0.723	0.7212	0.7107	Yes
	Original	Logistic	**0.6942**	0.6699	0.666	0.6693	0.6625	0.6658	0.6617	Yes
	Original	RFC	**0.612**	0.6048	0.6062	0.6087	0.602	0.591	0.0539	Yes
	PCA	Logistic	**0.6843**	0.6538	0.6566	0.656	0.6494	0.6481	0.6483	Yes
	PCA	RFC	0.3353	0.4273	0.4081	0.4222	**0.4393**	0.3616	-0.0002	Yes
PR-AUC										
	Autoencoder	Logistic	**0.7741**	0.7677	0.7711	0.7695	0.769	0.767	0.7686	Yes
	Autoencoder	RFC	**0.7771**	0.7618	0.7614	0.7595	0.7618	0.7629	0.7535	Yes
	Original	Logistic	**0.7315**	0.7106	0.7071	0.7097	0.7047	0.7067	0.7065	Yes
	Original	RFC	**0.6957**	0.6644	0.6662	0.6683	0.6569	0.6538	0.5416	Yes
	PCA	Logistic	**0.7231**	0.6965	0.6992	0.6982	0.6927	0.6913	0.6935	Yes
	PCA	RFC	**0.5833**	0.575	0.5592	0.5701	0.552	0.5359	0.5336	Yes
Precision										
	Autoencoder	Logistic	0.7956	0.7896	0.7924	0.7891	0.7936	0.8002	**0.8012**	No
	Autoencoder	RFC	**0.8464**	0.8174	0.8167	0.8095	0.807	0.8392	0.8314	Yes
	Original	Logistic	0.7305	0.7293	0.7243	0.7228	0.7336	0.7167	**0.7622**	Yes
	Original	RFC	**0.8942**	0.7711	0.7731	0.7777	0.7365	0.7708	0.19	Yes
	PCA	Logistic	0.7222	0.7058	0.7115	0.7046	0.708	0.6983	**0.7381**	Yes
	PCA	RFC	**0.9297**	0.8388	0.8109	0.8355	0.7544	0.7842	0	Yes
Recall										
	Autoencoder	Logistic	**0.7321**	0.7247	0.7291	0.7292	0.7232	0.7118	0.7139	Yes
	Autoencoder	RFC	0.6834	0.6817	0.6817	0.6851	**0.6928**	0.6603	0.6485	Yes
	Original	Logistic	**0.7099**	0.666	0.6638	0.671	0.6484	0.6712	0.6215	Yes
	Original	RFC	0.455	0.5205	0.5223	0.5219	**0.5419**	0.498	0.0169	Yes
	PCA	Logistic	**0.7006**	0.6606	0.6604	0.6657	0.6502	0.6577	0.6192	Yes
	PCA	RFC	0.1404	0.2536	0.2389	0.2464	**0.2955**	0.1956	0	Yes

Deep-ExtSMOTE (Dirichlet ExtSMOTE + Autoencoder) demonstrates superior performance across most of the evaluated metrics, including F1 Score, MCC, PR-AUC, and Recall, as evidenced by the bolded and blue-highlighted values in the comparison table. This approach consistently outperforms other scenarios, with the performance improvements over the baseline being statistically significant.

While the baseline Autoencoder achieves better precision compared to Deep-ExtSMOTE, this difference is not statistically significant. The robust performance of Deep-ExtSMOTE across multiple metrics highlights its effectiveness in addressing class imbalance and handling high-dimensional data, providing a more comprehensive solution compared to traditional methods. Furthermore, the increased reliability of Deep-ExtSMOTE as the

ratio of features to instances grows demonstrates its potential for effective application in a wide range of settings.

6.5 Discussion

Class imbalance remains a critical challenge in machine learning, particularly within high-dimensional datasets where the minority class is often underrepresented. Traditional methods, such as the original Synthetic Minority Over-sampling Technique (SMOTE), have been widely used to mitigate these issues by generating synthetic samples. However, SMOTE and its variants have limitations, especially in high-dimensional spaces where the risk of generating overlapping or less informative synthetic samples increases. Chawla et al. (2002) introduced SMOTE as a technique to address class imbalance by generating synthetic examples, but its effectiveness diminishes in the face of high-dimensional data, where the distribution of features becomes more complex and sparse.

Over the years, numerous advancements to SMOTE have been proposed to enhance its performance. For instance, Borderline-SMOTE (Han et al.; 2005) and Adaptive Synthetic Sampling (ADASYN) (He et al.; 2008) have improved the quality of synthetic samples by concentrating on regions near the decision boundary, where classification is most challenging. These methods, however, still face challenges in high-dimensional spaces, where the complexity of the data can lead to less effective sampling. The need for more sophisticated approaches that can handle the intricacies of high-dimensional data is evident.

The use of autoencoders for dimensionality reduction has been demonstrated as an effective approach to preserving important data characteristics while reducing computational complexity (Hinton and Salakhutdinov; 2006). Autoencoders compress data into a latent space, where the key features are retained, allowing models to focus on the most informative aspects of the data. This approach is particularly beneficial in high-dimensional datasets, where irrelevant or redundant features can obscure meaningful patterns. By focusing on the most prominent features, autoencoders help to alleviate the curse of dimensionality, thereby improving the performance of machine learning models.

Deep-ExtSMOTE represents a significant advancement in addressing class imbalance within high-dimensional datasets. By integrating autoencoder-based feature reduction with an enhanced version of SMOTE, Deep-ExtSMOTE seeks to improve the quality and effectiveness of synthetic sample generation, particularly in complex data environments.

This method enhances performance by leveraging the unique characteristics of Dirichlet SMOTE (Matharaarachchi et al.; 2024), particularly its ability to generate synthetic samples that better reflect the underlying distribution of the minority class. Unlike traditional SMOTE, Dirichlet SMOTE employs a Dirichlet distribution to assign weights during the interpolation process, allowing for more nuanced control over sample generation. This results in synthetic samples that are not only more representative of the minority class but also more evenly distributed within the feature space. By integrating this approach with autoencoders (Goodfellow et al.; 2016) for dimensionality reduction, Deep-ExtSMOTE simplifies the feature space while preserving the complex relationships within the data. This combination enables the generation of higher-quality synthetic samples that enhance the representativeness and balance of the minority class, leading to improved classification performance, particularly in high-dimensional and imbalanced datasets.

While there was a slight decrease in performance in high-dimensional settings, Deep-ExtSMOTE still demonstrates clear improvements over traditional SMOTE and its extensions, as evidenced in both synthetic and experimental studies. The strength of Deep-ExtSMOTE lies in its dual approach: the autoencoder component reduces data dimensionality, mitigating the curse of dimensionality, while the extended SMOTE mechanism refines synthetic sample generation through weighted averages of neighboring instances. This enhances the representativeness of minority class samples and improves overall classification performance.

Integrating autoencoder-based feature reduction into the SMOTE framework leads to more accurate synthetic data generation, better capturing the underlying data distribution and resulting in superior classifier performance. This enhancement is particularly valuable in high-dimensional scenarios where traditional methods often struggle.

In critical applications where accurately identifying minority class instances is crucial, such as medical diagnosis or fraud detection, even small improvements in recall or F1 score might justify the use of SMOTE or its extensions, as they could help capture cases that might otherwise be missed.

One limitation of this study is the absence of hyperparameter optimization for autoencoders in our experiments. Given the critical role hyperparameters play in the performance of autoencoders, their optimization could potentially lead to further improvements in classification outcomes. Future work should systematically explore and optimize these parameters to fully unlock the potential of Deep-ExtSMOTE.

Moreover, while our results are promising, further validation on diverse datasets and real-world applications is necessary to confirm the generalizability of Deep-ExtSMOTE.

Evaluating its performance across different domains and comparing it with other state-of-the-art methods would provide a more comprehensive understanding of its strengths and limitations.

6.5.1 Conclusion

In conclusion, Deep-ExtSMOTE represents a significant advancement in handling class imbalance, particularly in complex and high-dimensional datasets. By integrating autoencoders for dimensionality reduction with the Extended SMOTE technique, Deep-ExtSMOTE offers a more effective and robust approach to synthetic sample generation. This improvement is especially evident when compared to traditional SMOTE methods and their extensions. Future research should focus on optimizing hyperparameters and validating the approach across a broader range of applications to further establish its efficacy and utility.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016). Tensorflow: A system for large-scale machine learning, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283.

Breiman, L. (2001). Random forests, *Machine learning* **45**(1): 5–32.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: identifying density-based local outliers, *International Conference on Management of Data: Proceedings of the 2000 ACM SIGMOD international conference on Management of data : Dallas, Texas, United States; 15-18 May 2000*, ACM, New York, NY, USA, pp. 93–104.

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

Chollet, F. (2018). Keras: The python deep learning library, `https://keras.io`. Astrophysics Source Code Library, ascl:1806.022.

Cole, R. and Fanty, M. (1994). ISOLET, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C51G69.

Duan, L., Xu, L., Guo, F., Lee, J. and Yan, B. (2007). A local-density based spatial clustering algorithm with noise, *Information systems (Oxford)* **32**(7): 978–986.

Garzon, M. (2022). *Dimensionality reduction in data science*, Springer, Cham, Switzerland.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning*, Adaptive computation and machine learning, The MIT Press, Cambridge, Massachusetts.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning, *Advances in Intelligent Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 878–887.

He, H., Bai, Y., Garcia, E. and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning, *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Vol. 10, IEEE, pp. 1322–1328.

Hinton, G. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks, *Science (American Association for the Advancement of Science)* **313**(5786): 504–507.

Huynh, P.-H., Nguyen, V. H. and Do, T.-N. (2020). Improvements in the large p, small n classification issue, *SN computer science* **1**(4): 207–.

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

Matharaarachchi, S., Domaratzki, M. and Muthukumarana, S. (2024). Enhancing smote for imbalanced data with abnormal minority instances, *Machine learning with applications* **18**: 100597–.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

Salekshahrezaee, Z., Leevy, J. L. and Khoshgoftaar, T. M. (2021). Feature extraction for class imbalance using a convolutional autoencoder and data sampling, *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, pp. 217–223.

Suja, A. A. and Nayahi, J. J. V. (2023). Classification of imbalanced data using SMOTE and autoencoder based deep convolutional neural network, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **31**: 437–469.

Weisberg, S. (2005). *Applied Linear Regression*, John Wiley and Sons, Incorporated, Hoboken.

Zhang, C. and Bom, S. (2021). Auto-encoder based model for high-dimensional imbalanced industrial data, *Neural Information Processing*, Communications in Computer and Information Science, Springer International Publishing, Cham, pp. 265–273.

Chapter 7

Discussion

The research presented in this thesis addresses critical challenges in data science, particularly in the context of class imbalance and high-dimensional data, with a specific focus on the presence of abnormal instances in the minority class. Motivated by the significant public health implications of long COVID, this work begins with the identification of symptom patterns using innovative data mining techniques and progresses to the application of machine learning for the prediction of long COVID cases. To support these efforts, novel methodologies were developed to tackle the inherent difficulties posed by class imbalance in complex datasets. This discussion synthesizes the findings across multiple chapters, illustrating how each methodological advancement contributes to the overarching goal of improving predictive accuracy in challenging data environments.

Chapters 3 and 4 of this thesis focus on the identification and prediction of long COVID cases by using different datasets and methodologies. Chapter 3 explores the identification of long COVID symptom patterns through the use of Association Rule Mining and Sentiment Analysis applied to social media data. This approach was particularly innovative, as it was developed at a time when long COVID-related data were not readily available. The findings in this chapter lay the essential groundwork for understanding the symptoms of long COVID, which is crucial for developing robust predictive models. Building upon these insights, Chapter 4 applies machine learning techniques to clinical notes data to predict long COVID cases specifically in Manitoba. A significant challenge addressed in this chapter is the class imbalance present in the dataset, which is critical to ensuring that minority class instances, such as those representing long COVID cases, are accurately predicted without the model biasing toward the majority class. The connection between these chapters is the progression from identifying relevant patterns in symptom data to applying these patterns in a predictive model to enhance the accuracy of long COVID diagnoses.

Subsequently, Chapters 5 and 6 introduce and develop novel methodologies designed to address the class imbalance issues that are pivotal for improving the models discussed in Chapter 4, particularly in complex scenarios. Chapter 5 presents extensions of the SMOTE technique aimed at handling abnormal instances within imbalanced datasets, thereby enhancing classification performance. These extensions are crucial in dealing with the intricacies of real-world data, where abnormalities and outliers can significantly affect model accuracy. Chapter 6 takes these advancements further by integrating autoencoders with the advanced SMOTE techniques discussed in Chapter 5. This integration is particularly beneficial for addressing class imbalance in high-dimensional data, where the curse of dimensionality often exacerbates the imbalance problem. The methodologies developed in these chapters are not just theoretical contributions; they are directly applicable to the predictive tasks outlined in Chapters 3 and 4, providing enhanced tools to effectively manage the complexities associated with imbalanced and high-dimensional datasets.

7.1 Summary of Contributions and Novel Developments

In this thesis, I have focused on addressing the significant challenge of class imbalance in high-dimensional data, particularly in classification tasks (Krawczyk; 2016). This issue is especially critical in fields such as medical diagnostics (Mazurowski et al.; 2008), fraud detection (Yang et al.; 2009), and churn prediction (Zhu et al.; 2018), where the minority class often represents rare but vital cases. Traditional machine learning algorithms typically struggle to detect these minority instances, exacerbating the need for novel techniques to mitigate this bias. The problem becomes even more challenging in high-dimensional data or when outliers are present within the minority class.

While Sáez et al. (2015) recommend removing noisy or outlier samples to improve model performance, this approach can be risky, particularly in imbalanced datasets where the minority group has limited data. Reducing this already scarce data may lead to the loss of critical information, weakening the model's ability to generalize. Distinguishing between true outliers and legitimate minority class representatives is challenging; removing key variations could result in the misclassification of future instances (Krawczyk; 2016). In high-stakes applications like medical diagnosis or fraud detection, careful consideration is essential, as improper removal could compromise the model's accuracy and effectiveness, with high dimensionality further exacerbating these challenges.

This research introduces several advanced resampling methods that not only address class imbalance but also effectively handle abnormal instances, such as outliers or noisy data points within the minority class, while also considering the high-dimensional aspect. Below is a summary of the novel proposed methods (Matharaarachchi, Domaratzki and Muthukumarana; 2024; Matharaarachchi, Domaratzki, and Muthukumarana; 2024).

1. **Distance-ExtSMOTE:** Distance-ExtSMOTE is an advanced extension of the Synthetic Minority Over-sampling Technique (SMOTE) that leverages the concept of distance-based weighting to generate synthetic samples. Unlike traditional SMOTE, which blindly generates synthetic instances by interpolating between minority class instances, Distance-ExtSMOTE incorporates a weighted average of neighboring instances. The weights are determined by the inverse distance between each neighboring point and the median centroid of the minority class. This ensures that synthetic samples are generated closer to the decision boundary, effectively preserving minority class characteristics. Additionally, Distance-ExtSMOTE handles abnormal instances by incorporating a robustness factor that reduces the influence of outliers during the sample generation process. This approach results in more reliable synthetic data.

2. **Dirichlet-ExtSMOTE:** Dirichlet-ExtSMOTE introduces a novel sampling approach using the Dirichlet distribution to generate synthetic minority samples. By leveraging this probabilistic model, Dirichlet-ExtSMOTE provides controlled and diverse synthetic sample generation, especially in highly imbalanced datasets. The Dirichlet distribution is used to assign weights, with parameters influenced by distances to neighbors, offering a flexible and probabilistic approach to weight assignment. This method is particularly effective in managing abnormal instances, as it allows for control over variability and ensures that outliers have less influence on the model, resulting in a more accurate representation of the minority class.

3. **FCRP SMOTE:** The FCRP-SMOTE method employs the principles of the Finite Chinese Restaurant Process to generate synthetic samples in a manner that preserves the local structure of the minority class. In this method, distances influence initial probabilities, and weights are adjusted iteratively to capture evolving data patterns and local data characteristics dynamically. This clustering-based mechanism assigns synthetic samples to clusters based on the size of the clusters, similar to seating arrangements in a Chinese restaurant. This approach is particularly adept at handling abnormal instances by assigning lower probabilities to clusters with outliers, thereby reducing their influence on the synthetic sample generation process. FCRP-SMOTE is effective in datasets where the minority

class has complex and overlapping distributions with the majority class, and where abnormal instances could otherwise skew the classification results.

4. **BGMM SMOTE:** BGMM-SMOTE incorporates Bayesian Gaussian Mixture Models to generate synthetic samples, leveraging the posterior probabilities of cluster memberships. This method models the minority class distribution as a mixture of Gaussians, with cluster probabilities used to assign weights that reflect the likelihood of instances belonging to specific clusters. Specifically, the weights are determined by the probabilities of the cluster where the median centroid of the minority class belongs to, effectively capturing complex data distributions. This approach provides a robust framework for generating synthetic data even in the presence of abnormal instances. The Bayesian framework also allows for the integration of uncertainty, ensuring that outliers and noisy data points are less likely to influence the generation process. This results in more representative synthetic samples that enhance the classifier's ability to detect minority class instances, particularly in datasets with subgroups within the minority class.

5. **Deep-ExtSMOTE:** DeepExt-SMOTE extends the concept of synthetic sample generation by integrating autoencoders with the previously proposed four methods (e.g., Dirichlet-ExtSMOTE). Autoencoders are used to learn a compressed, lower-dimensional representation of the data, capturing the essential features of the minority class. The classification is then performed directly in this latent space rather than mapping the synthetic samples back to the original high-dimensional space. This combination of autoencoders with advanced resampling techniques addresses the challenges of the curse of dimensionality, ensuring that the generated samples are both diverse and representative of the minority class.

7.1.1 Mitigate the Risks Associated with Rebalancing Imbalanced Datasets

Rebalancing is essential for addressing class imbalance in datasets, but it can pose risks if not implemented carefully. The process involves modifying the class distribution to create a more balanced dataset, often through techniques like oversampling, undersampling, or synthetic data generation. Each of these methods has potential pitfalls that can compromise the effectiveness and reliability of predictive models. In the following sections, we explore these challenges (Krawczyk; 2016; Niaz et al.; 2022; Fernández et al.; 2011; Fernandez et al.; 2018), and outline how our proposed methods mitigate them to enhance model performance and robustness.

Overfitting

One of the primary risks of rebalancing, particularly with oversampling techniques, is overfitting. By duplicating instances of the minority class, the model may learn the noise and specific patterns of these repeated instances rather than the underlying general trends. This can result in a model that performs well on the training data but fails to generalize to new, unseen data. Overfitting is particularly problematic in applications like fraud detection and disease diagnosis, where the ability to generalize to new cases is crucial.

Our proposed methods, including Dirichlet ExtSMOTE, mitigate overfitting by generating synthetic data points that accurately reflect the intrinsic structure of the minority class through a weighted average of neighbouring instances. This approach helps prevent the model from learning spurious patterns that do not generalize well to unseen data, as it avoids excessively amplifying noise or outliers. Dirichlet ExtSMOTE, in particular, ensures a balanced distribution of synthetic samples, reducing the risk of classifiers focusing too heavily on localized or anomalous regions of the feature space. Careful parameter tuning is essential to maintain diversity and accurately represent the minority class. Implementing cross-validation techniques allows for early detection of overfitting by assessing model performance on validation datasets, facilitating necessary adjustments to the sampling process. Additionally, combining Dirichlet ExtSMOTE with ensemble learning methods, such as Random Forests, further reduces overfitting by aggregating predictions across multiple models, diminishing the impact of any single overfitted model. Employing a hybrid approach that integrates Dirichlet ExtSMOTE with preprocessing techniques like feature selection or dimensionality reduction enhances model robustness, minimizing overfitting risks and ensuring better generalization to unseen data.

Loss of Information

Undersampling involves reducing the number of instances in the majority class to balance the dataset. While this can help in creating a balanced class distribution, it comes at the cost of potentially losing valuable information. Reducing the majority class can lead to a less comprehensive understanding of the data, missing important patterns and trends. For example, in credit rate determination, undersampling non-defaulted loans can discard valuable information about creditworthiness.

Our proposed methods, aim to address the potential loss of information in imbalanced datasets through oversampling techniques. By generating synthetic samples that closely resemble the minority class's intrinsic structure, these methods help ensure that the

minority class is adequately represented without introducing excessive noise or artifacts. The weighted averaging approach used in these methods creates synthetic data points based on the proximity of neighboring instances, preserving important characteristics of the minority class while minimizing the risk of information loss. This careful synthesis of new data maintains the diversity and distribution of the minority class, allowing classifiers to better learn and generalize from the enriched dataset.

Sample Bias

The process of selecting which instances to duplicate or remove can introduce bias. If the selection process does not appropriately capture the diversity of the data, the resulting dataset may not be representative of the real-world scenario. This bias can skew the model's predictions and reduce its effectiveness. For instance, in churn prediction, oversampling specific patterns of customer behavior can lead to a biased model that does not generalize well across the entire customer base.

Our proposed methods are designed to mitigate sample bias in imbalanced datasets through advanced oversampling techniques. Sample bias occurs when the model learns to favor the majority class due to its overrepresentation, potentially leading to skewed predictions and poor generalization. By generating synthetic samples that accurately represent the minority class, these methods aim to balance the class distribution and reduce the inherent bias. The use of a weighted average of neighboring instances ensures that synthetic data points capture the true diversity of the minority class, preventing the model from being biased toward specific minority instances that may not be representative. Additionally, methods like Dirichlet ExtSMOTE employ a probabilistic weight determination mechanism using a Dirichlet distribution, which allows for a more flexible and dynamic generation of synthetic samples. This probabilistic approach enables the method to account for the inherent variability in the data, ensuring that synthetic samples are distributed in a manner that better reflects the true underlying distribution of the minority class. This approach helps in creating a more uniform and unbiased training dataset, allowing classifiers to learn decision boundaries that are fair and representative of the underlying data distribution. Moreover, the strategic design of these methods focuses on minimizing the introduction of noise and artifacts, further reducing the risk of sample bias and enhancing the overall robustness of machine learning models in handling imbalanced data.

Synthetic Instance Quality

Techniques like SMOTE generate synthetic instances by interpolating between existing minority class instances. The quality and realism of these synthetic instances can

vary, potentially introducing unrealistic or noisy data. Poor-quality synthetic data can mislead the model, resulting in inaccurate predictions. This is particularly critical in gene expression analysis and protein structure classification, where the accuracy of the generated data directly impacts the model's performance.

The proposed methods enhance the quality of synthetic instances by generating samples that accurately capture the diversity and distribution of the minority class. These methods use a weighted average of neighbouring instances, ensuring that synthetic data points reflect the true characteristics of the minority class. The probabilistic weight determination allows for dynamic and balanced sample generation, which prevents synthetic samples from being overly clustered or excessively sparse. This approach minimizes the risk of introducing noise or artifacts, thereby maintaining the integrity of the synthetic data. By filling gaps within the minority class distribution, these methods reduce overfitting and improve model generalization, ultimately providing high-quality synthetic instances that enhance model training and performance on imbalanced datasets.

Handling Abnormal Instances

Another challenge is the presence of abnormal instances or outliers within the minority class. These instances can significantly affect the performance of rebalancing techniques. Oversampling may amplify the impact of outliers, leading to models that overfit these abnormal instances. Conversely, undersampling may inadvertently remove these important but rare instances, reducing the model's ability to detect them. In natural disaster prediction and threat detection from surveillance videos, accurately identifying and handling outliers is crucial for effective predictions.

One of the major contributions of our proposed methods is their ability to effectively handle abnormal instances, such as outliers, within the minority class. The methods proposed in this thesis address this issue by employing advanced weighting mechanisms and probabilistic approaches to minimize the impact of outliers during the generation of synthetic samples. By assigning lower weights to outliers and focusing on the more representative instances within the minority class, these methods create synthetic data that better reflects the true underlying distribution, enhancing the quality and reliability of the training dataset.

Calibration Issue

SMOTE can introduce calibration issues (Foster and Vohra; 1998) in models by altering the class distribution through synthetic data generation. By interpolating between

minority class instances, SMOTE can lead to poorly calibrated models, as the synthetic samples may not fully represent the true distribution of the minority class. This can result in predictions that are either overly confident or too conservative, affecting the model's ability to estimate probabilities accurately.

In our analysis, we address the calibration issues associated with SMOTE-generated synthetic data by applying isotonic regression as a post-training calibration method (Foster and Vohra; 1998; Zadrozny and Elkan; 2001; Pedregosa et al.; 2011). Isotonic regression is a non-parametric technique that adjusts the predicted probabilities to better align with the actual outcomes, ensuring that the model's confidence levels are accurate and reliable. By recalibrating the model's predictions, isotonic regression helps correct any distortions introduced by the synthetic samples, thus improving the model's ability to provide well-calibrated probability estimates. This step is particularly crucial in applications where accurate probability assessments are necessary for informed decision-making, such as in healthcare diagnostics and financial risk evaluations. By incorporating isotonic regression, we enhance the interpretability and trustworthiness of our models, ensuring that they are not only effective but also reliable in real-world scenarios.

Rebalancing imbalanced data is a double-edged sword. While it is necessary for improving the performance of predictive models, it also poses several risks if not done correctly. Our proposed methods—Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE, and BGMM SMOTE—are designed to address these challenges effectively. They mitigate risks such as overfitting by using advanced interpolation techniques and probabilistic weighting to generate high-quality synthetic instances. These methods also focus on preserving important information and reducing sample bias, ensuring that the synthetic data accurately reflects the minority class distribution. Additionally, they handle abnormal instances by minimizing the impact of outliers, and they improve calibration through post-training methods like isotonic regression. By understanding these potential pitfalls and applying our advanced techniques with careful tuning, we can enhance model accuracy and robustness. This approach allows practitioners to develop effective predictive models that perform well across diverse applications, from finance and biomedicine to security management, industrial production, and information technology.

7.2 Overall Impact and Future Directions

The novel methods developed in this thesis represent significant advancements in addressing class imbalance and handling abnormal instances in high-dimensional data.

These methods have been designed to tackle challenges associated with imbalanced datasets, including preserving local structures, managing complex distributions, and mitigating the impact of abnormal instances. By integrating probabilistic models, clustering mechanisms, and deep learning techniques, this work offers a comprehensive toolkit for researchers and practitioners facing the complexities of imbalanced data.

While these contributions have primarily focused on binary classification, future research could expand these methods to address the unique challenges of multi-class classification (Rachmatullah; 2022). Multi-class scenarios introduce additional complexities, such as varying degrees of imbalance across multiple classes and the interactions between minority classes. Enhancing these methods to effectively manage multi-class imbalances would be a critical step forward in creating more robust and generalizable models.

Moreover, combining these methods with advanced techniques such as transfer learning or ensemble approaches (Kumaran et al.; 2024) could further enhance their effectiveness. Transfer learning could be particularly useful in leveraging knowledge from related tasks or domains to improve classification performance in underrepresented classes. Ensemble methods, on the other hand, could integrate the strengths of different classifiers to create a more robust and accurate model, especially in complex multi-class settings.

Bibliography

Fernández, A., García, S. and Herrera, F. (2011). Addressing the classification with imbalanced data: Open problems and new challenges on class distribution, *Hybrid Artificial Intelligent Systems*, Vol. 6678 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–10.

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

Foster, D. P. and Vohra, R. V. (1998). Asymptotic calibration, *Biometrika* **85**(2): 379–390.

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

Kumaran, S. Y., Jeya, J. J., Mahesh, T. R., Khan, S. B., Alzahrani, S. and Alojail, M. (2024). Explainable lung cancer classification with ensemble transfer learning of vgg16, resnet50 and inceptionv3 using grad-cam, *BMC medical imaging* **24**(1): 176–19.

Matharaarachchi, S., Domaratzki, M., and Muthukumarana, S. (2024). DeepExtSMOTE: Integrating autoencoders for advanced mitigation of class imbalance in high-dimensional data classification (to be submitted).

Matharaarachchi, S., Domaratzki, M. and Muthukumarana, S. (2024). Enhancing smote for imbalanced data with abnormal minority instances, *Machine learning with applications* **18**: 100597–.

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A. and Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance, *Neural networks : the official journal of the International Neural Network Society* **21**(2-3): 427—436.

Niaz, N. U., Shahariar, K. N. and Patwary, M. J. A. (2022). Class imbalance problems in machine learning: A review of methods and future challenges, *Proceedings of the 2nd International Conference on Computing Advancements*, ACM, New York, NY, USA, pp. 485–490.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

Rachmatullah, M. (2022). The application of repeated smote for multi class classification on imbalanced data, *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer* **22**.

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.

Yang, Z., Tang, W., Shintemirov, A. and Wu, Q. (2009). Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **39**: 597 – 610.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown, *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 204–213.

Zhu, B., Baesens, B., Backiel, A. and vanden Broucke, S. K. L. M. (2018). Benchmarking sampling techniques for imbalance learning in churn prediction, *Journal of the Operational Research Society* **69**(1): 49–65.

Appendices

Appendix A

A.1 Assessing Performance Across Different Dataset Shapes

To evaluate performance across various dataset shapes we used the 'make_moons' and 'make_circles' functions from the Python scikit-learn library to generate data (Pedregosa et al.; 2011) synthetically. The visual representation of the resulting outcomes, inclusive of an additional outlier, is presented in Fig. A.1 and Fig. A.2, showcasing the final results for each dataset shape.

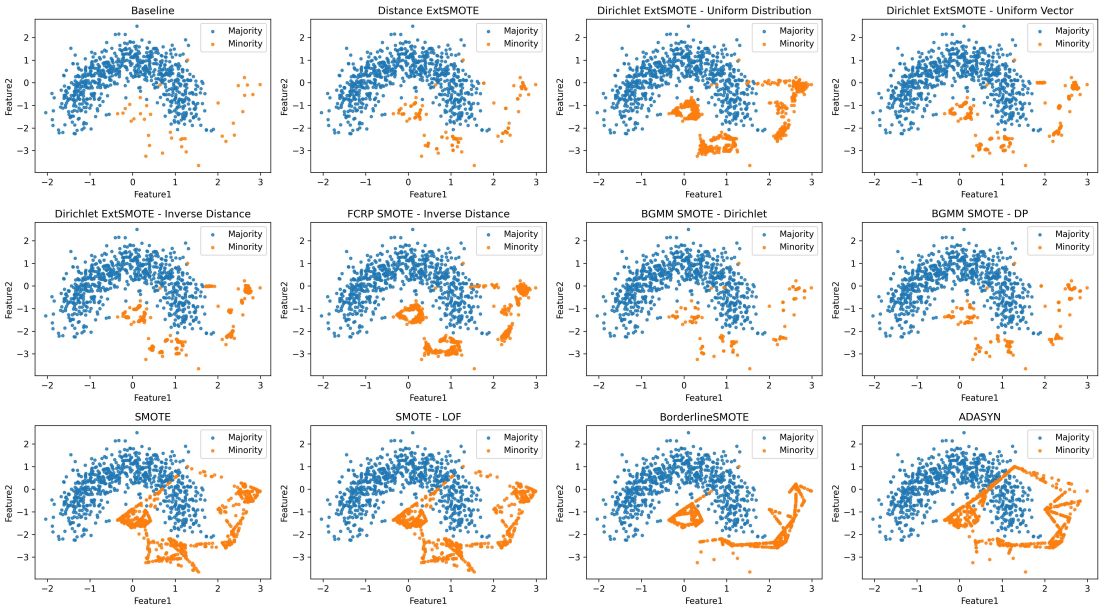


Figure A.1: Results comparison with noisy moons dataset with one abnormal instance when noise = 0.2

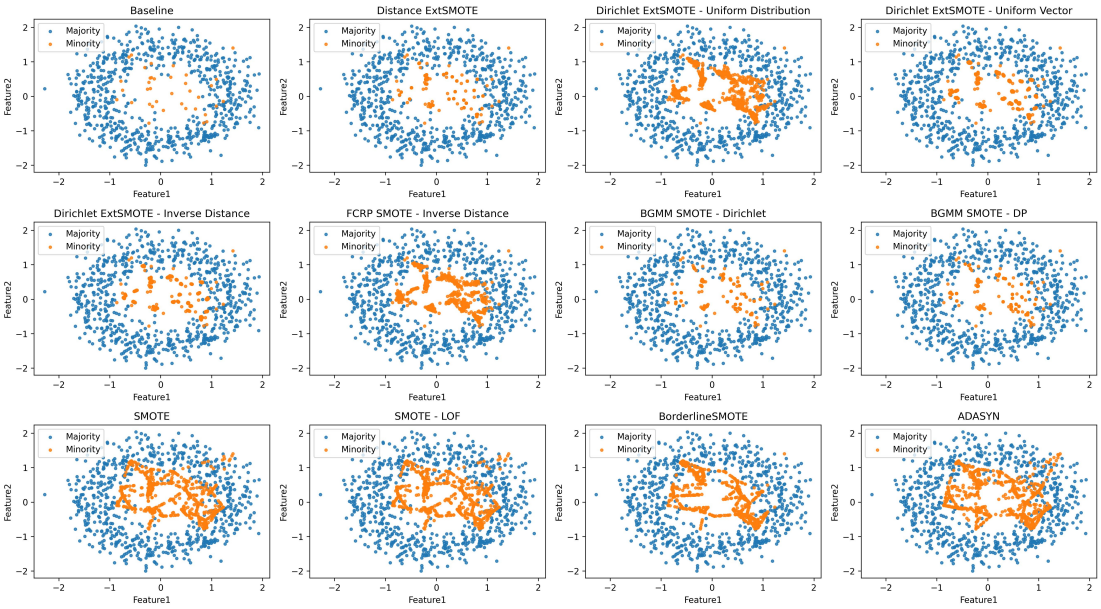


Figure A.2: Results comparison with noisy circles dataset with one abnormal instance when noise = 0.2.

A.2 Simulation Results for Different Accuracy Measures

The box plots in Fig. A.3, Fig. A.4, and Fig. A.5 show the distribution of the F1 scores across 100 simulated datasets, each with two different imbalance ratios and an outlier ratio of 0.1. These F1 scores were computed for three different classifiers: Logistic Regression, k-Nearest Neighbors (k-NN), and Random Forest. These visual representations provide a clear comparison of the performance variability and central tendency of each classifier under the specified conditions.

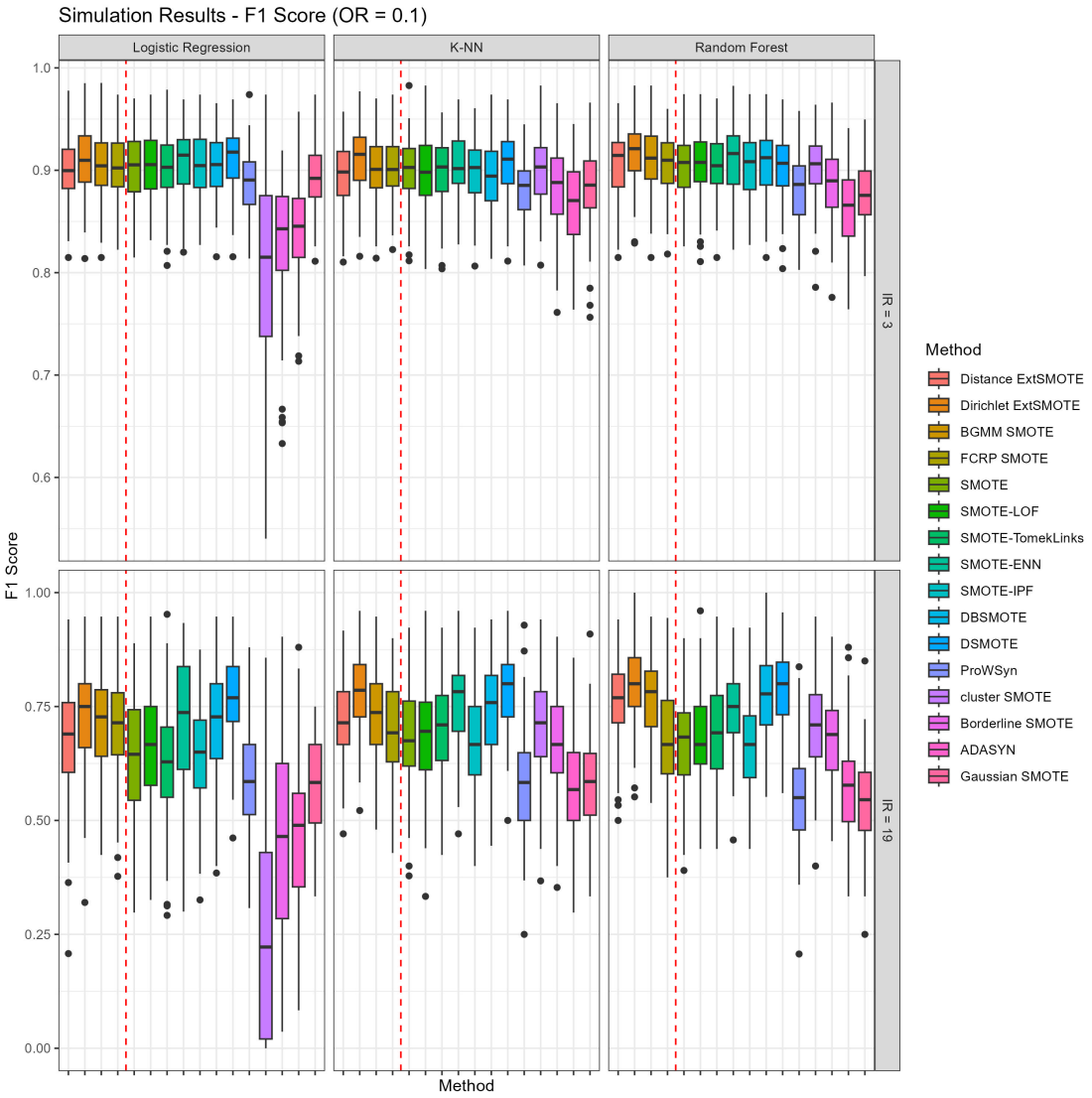


Figure A.3: F1 Scores across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

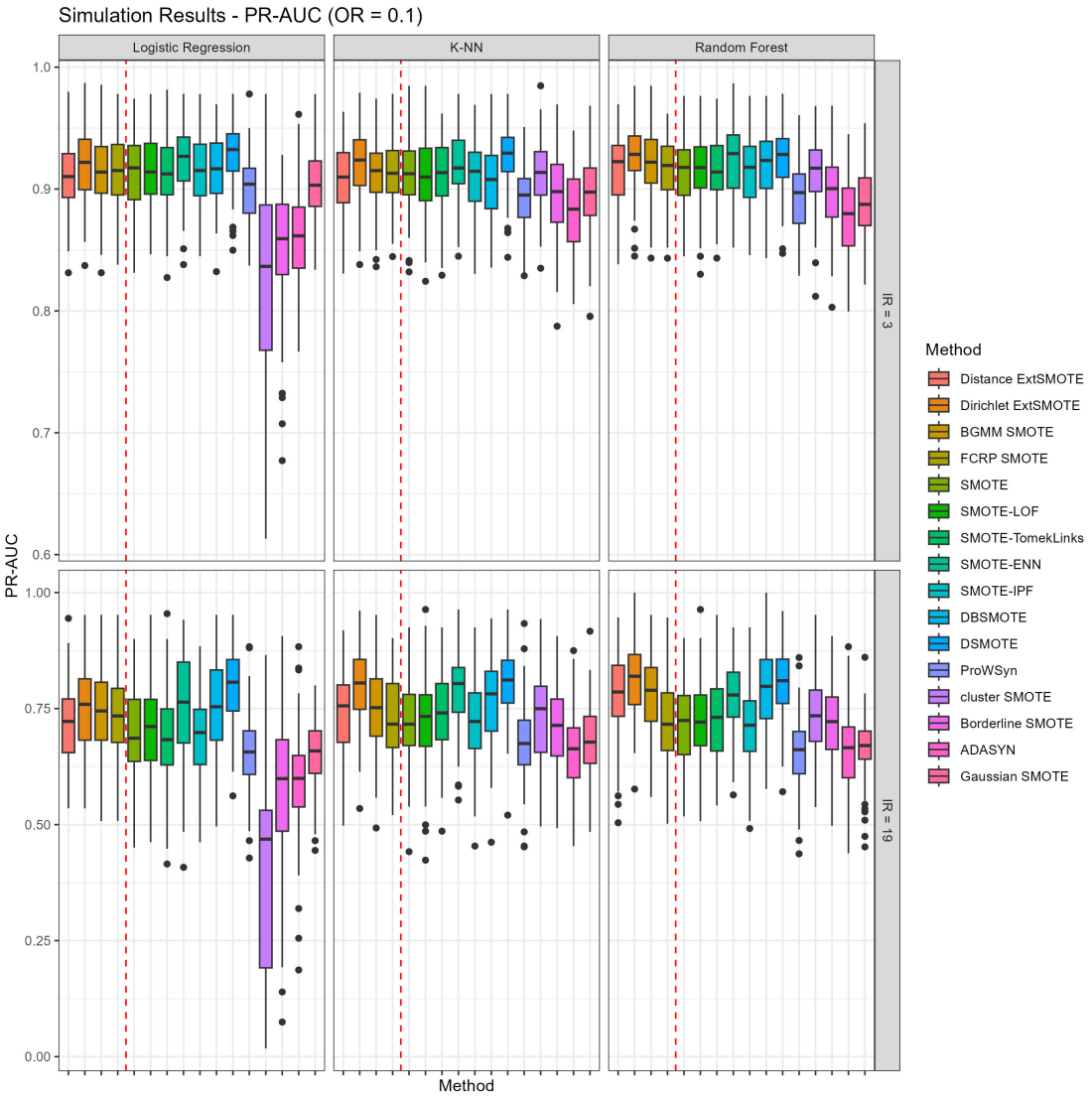


Figure A.4: PR-AUCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

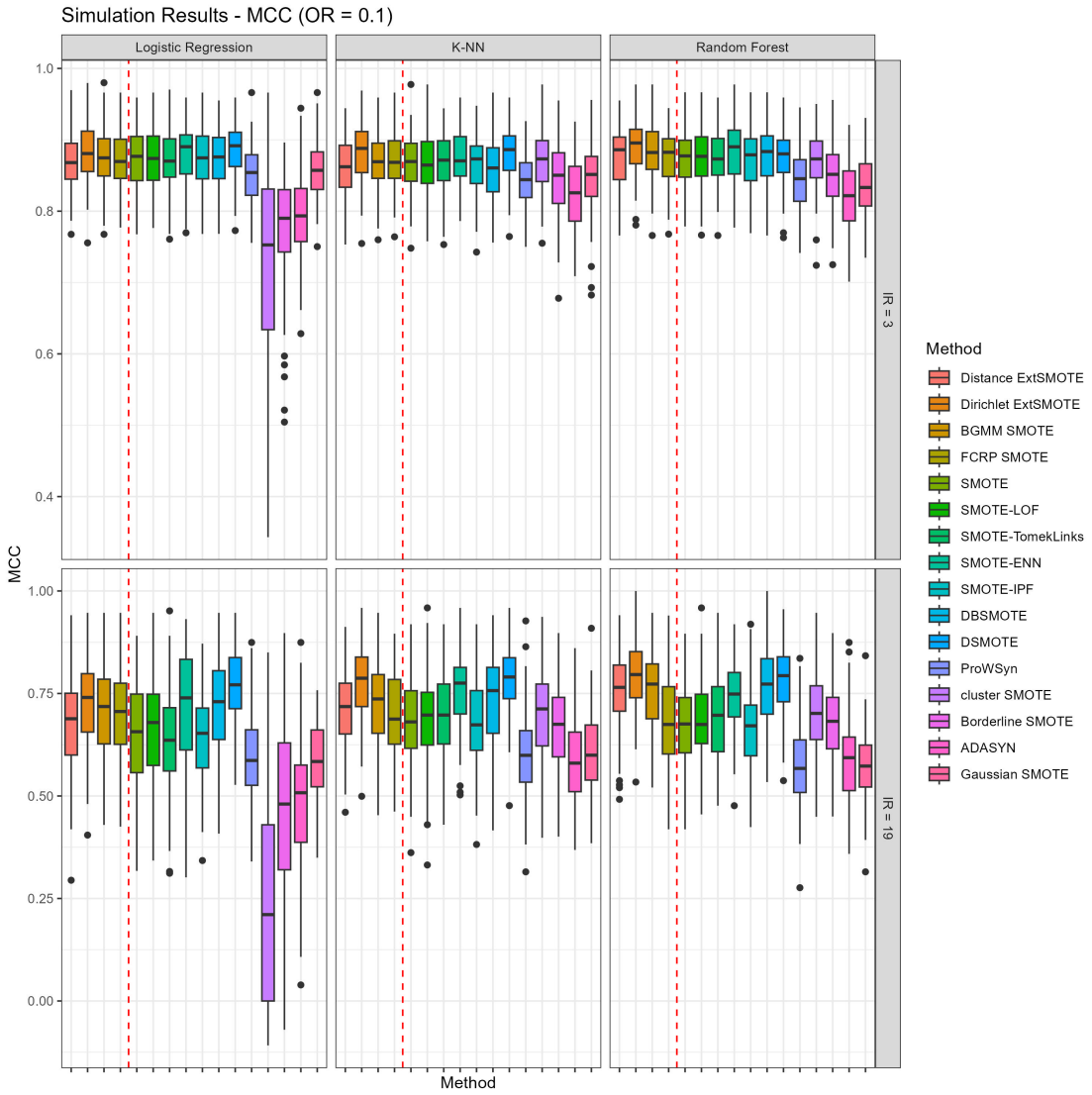


Figure A.5: MCCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

A.3 Sensitivity Analysis of the Parameters

Sensitivity analysis was conducted to evaluate the impact of various parameter settings on the performance of the proposed methods. This provided insights into the stability of our methods, ensuring that they remain effective across a range of parameter values and dataset characteristics.

A.3.1 Parameters of Dirichlet ExtSMOTE

A.3.1.1 Impact of Different Multipliers (m)

Expected Values with Multipliers

Given a vector of parameters $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_k]$, the expected value for each weight w_i in the Dirichlet distribution $Dir(\alpha)$ is (Ng et al.; 2011):

$$E[w_i] = \frac{\alpha_i}{\sum_{j=1}^{k} \alpha_j}$$

Let the initial inverse distances be $d_1^{-1}, d_2^{-1}, \ldots, d_k^{-1}$. We define the parameters of the Dirichlet distribution as:

$$\alpha_i = m \cdot d_i^{-1}$$

Where m is a multiplier.

Substituting $\alpha_i = m \cdot d_i^{-1}$ into the expected value formula:

$$E[w_i] = \frac{m \cdot d_i^{-1}}{\sum_{j=1}^{k} m \cdot d_j^{-1}}$$

Thus, the expected values $E[w_i]$ are independent of the multiplier m and depend only on the relative proportions of the inverse distances.

Expected Values with Variability and Concentration

The variance of each weight w_i in the Dirichlet distribution is given by (Ng et al.; 2011):

$$Var(w_i) = \frac{\alpha_i(\sum_{j=1}^{k} \alpha_j - \alpha_i)}{(\sum_{j=1}^{k} \alpha_j)^2(\sum_{j=1}^{k} \alpha_j + 1)}$$

Substituting $\alpha_i = m \cdot d_i^{-1}$ into the expected value formula:

$$\sum_{j=1}^{k} \alpha_j = m \cdot \sum_{j=1}^{k} m \cdot d_j^{-1}$$

$$Var(w_i) = \frac{m \cdot d_i^{-1}(m \cdot \sum_{j=1}^{k} d_j^{-1} - m \cdot d_i^{-1})}{(m \cdot \sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)}$$

Simplifying:

$$\begin{aligned} Var(w_i) &= \frac{m \cdot d_i^{-1} \cdot m(\sum_{j=1}^{k} d_j^{-1} - d_i^{-1})}{m^2(\sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)} \\ &= \frac{m^2 \cdot d_i^{-1}(\sum_{j=1}^{k} d_j^{-1} - d_i^{-1})}{m^2(\sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)} \\ &= \frac{d_i^{-1}(\sum_{j=1}^{k} d_j^{-1} - d_i^{-1})}{(\sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)} \end{aligned}$$

From this expression, it is clear that as m increases, the term $(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)$ also increases, thereby reducing the overall variance. Conversely, for small values of m, this term does not significantly increase, leading to higher variance.

The expected values of the weights in the Dirichlet distribution remain constant across different multipliers because they depend on the relative proportions of the parameters. However, the variability of the weights differs significantly. Higher multipliers (e.g., 100) result in lower variability, leading to more stable and predictable synthetic samples. In contrast, lower multipliers (e.g., 0.01) introduce higher variability, producing a more diverse but less stable set of synthetic samples. This mathematical analysis underscores the importance of choosing an appropriate multiplier based on the desired balance between diversity and stability in synthetic sample generation for addressing class imbalance.

Fig.A.6 and Fig. A.7 present the mean F1 scores for different multipliers of Dirichlet ExtSMOTE, considering outlier ratios of 0.05 and 0.1 respectively, across two different imbalance ratios for each type. It is important to note that the F1 scores vary significantly based on the selection of the classifier, the multiplier, and the number of outliers in the dataset, especially in scenarios with highly imbalanced data. This underscores the critical need for hyperparameter optimization to achieve higher performance. The

figures illustrate how the careful tuning of parameters can lead to significant improvements in classifier effectiveness, demonstrating the necessity of customized approaches in handling imbalanced datasets with varying outlier ratios.

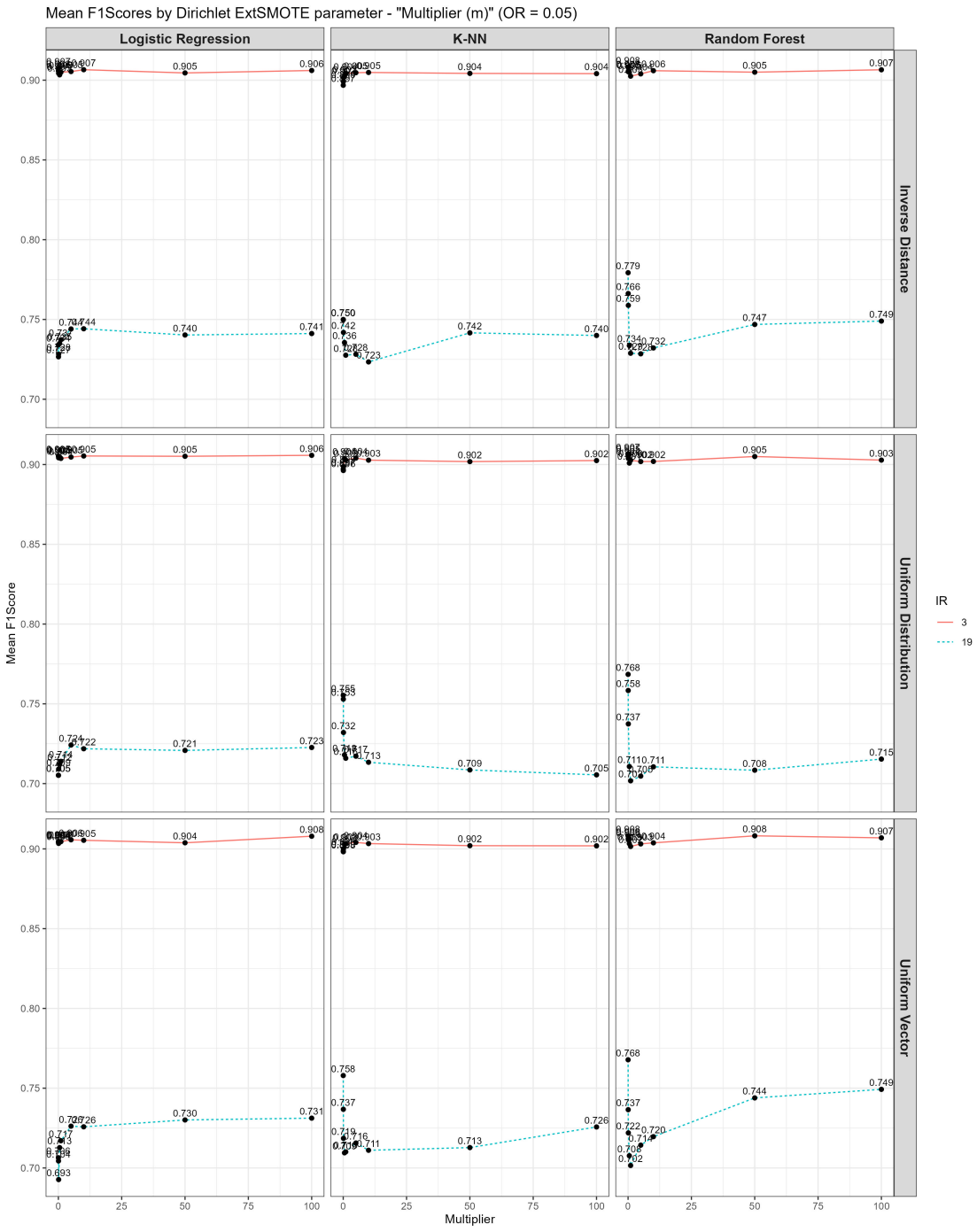


Figure A.6: Mean F1 Scores for different multipliers of Dirichlet ExtSMOTE for an outlier ratio of 0.05 and two different imbalance ratios for each type.

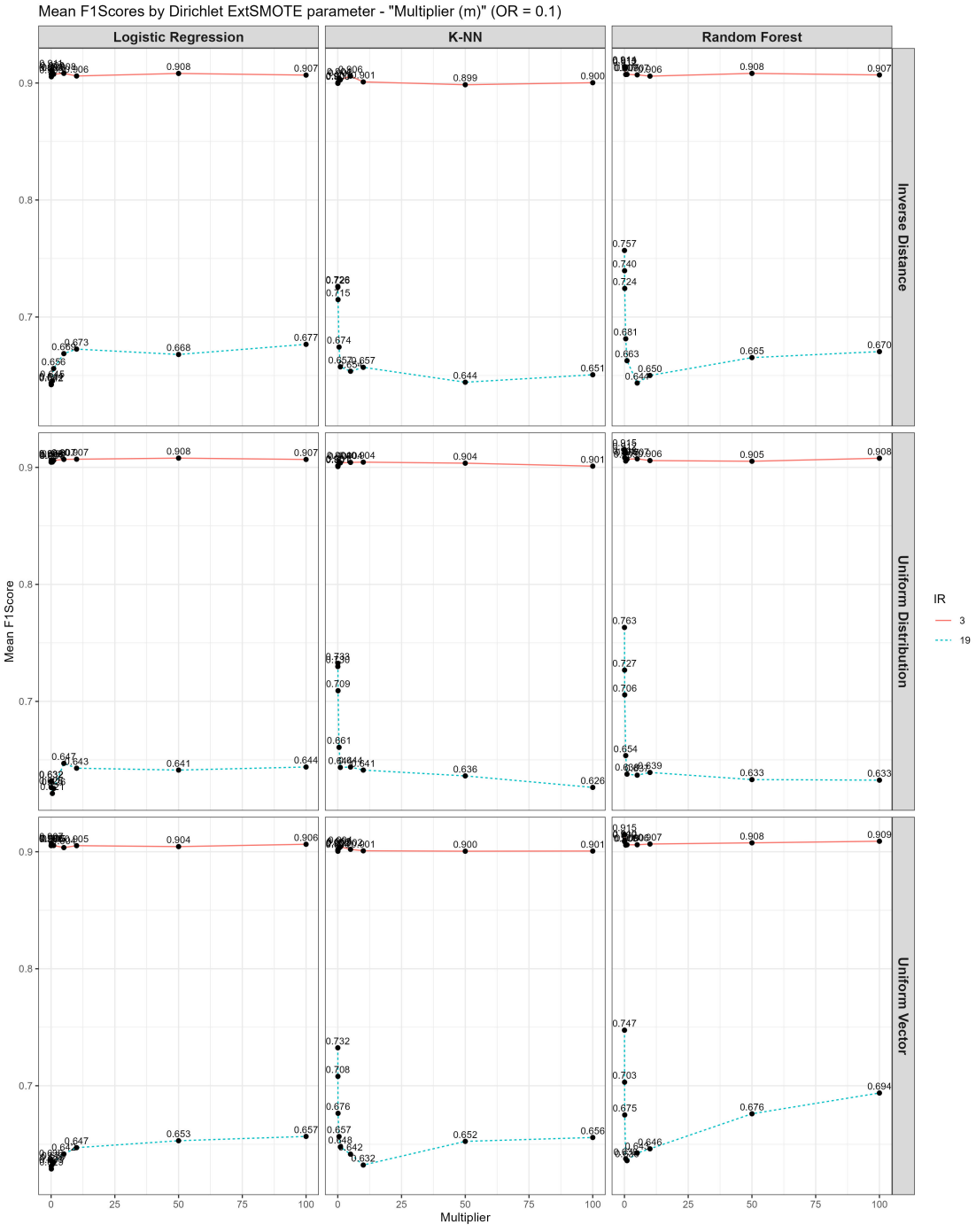


Figure A.7: Mean F1 Scores for different multipliers of Dirichlet ExtSMOTE for an outlier ratio of 0.1 and two different imbalance ratios for each type.

A.3.1.2 Impact of Different Types

Three different types are introduced in the Dirichlet ExtSMOTE method: Inverse Distance (D), Uniform Distribution (UD), and Uniform Vector (UV). Fig. A.8 illustrates the changes in Mean F1 Scores for these different types under two varying outlier ratios and two different imbalance ratios.

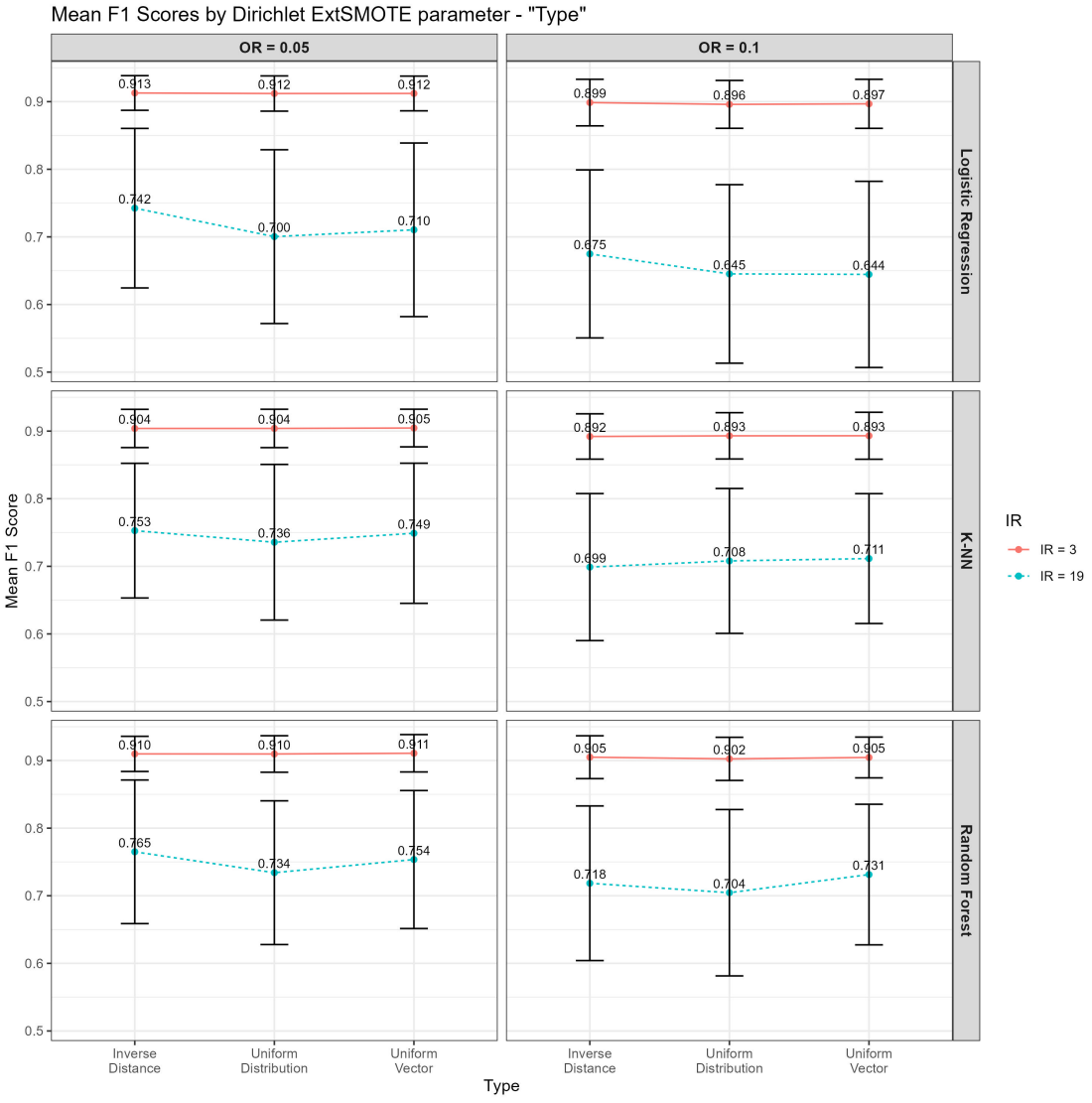


Figure A.8: Mean F1 Scores for different types of Dirichlet ExtSMOTE for two different outlier ratios and two different imbalance ratios.

A.3.2 Parameters of FCRP SMOTE

A.3.2.1 Impact of Different α Values

We evaluated the performance changes of FCRP SMOTE as the parameter alpha varied from 0.1 to 1. Figure A.9 shows the mean F1 scores for two imbalance rates and outlier ratios using three classifiers, highlighting the impact of alpha on the effectiveness of FCRP SMOTE under different conditions.

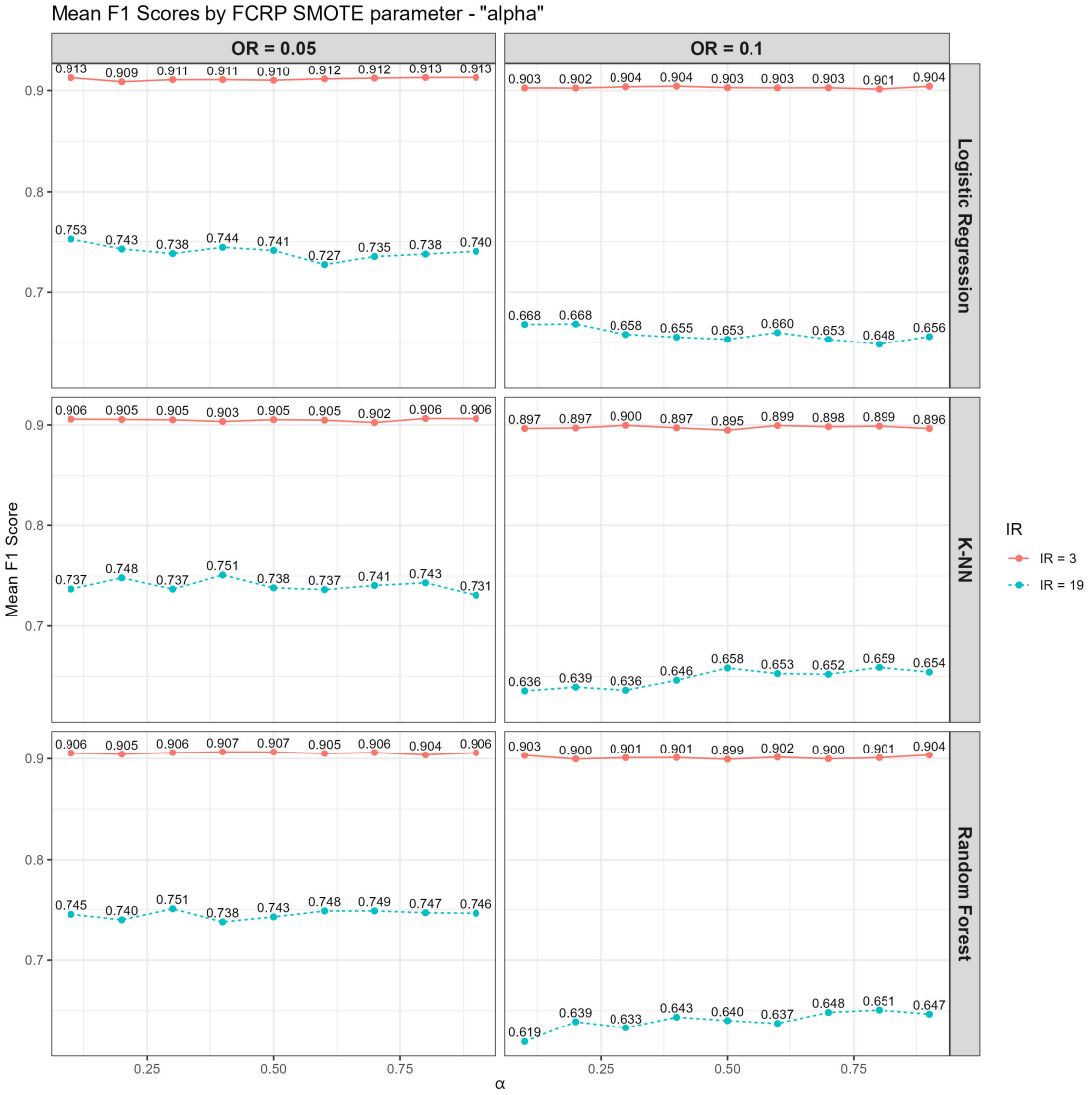


Figure A.9: Mean F1 Scores for different alpha values of FCRP SMOTE for two different outlier ratios and two different imbalance ratios.

A.3.3 Parameters of BGMM SMOTE

A.3.3.1 Impact of Different Priors

Two priors are introduced with BGMM SMOTE. Figure A.10 shows the mean F1 scores for two imbalance rates and outlier ratios using three classifiers, highlighting the impact of selecting the prior on the effectiveness of BGMM SMOTE under different conditions.

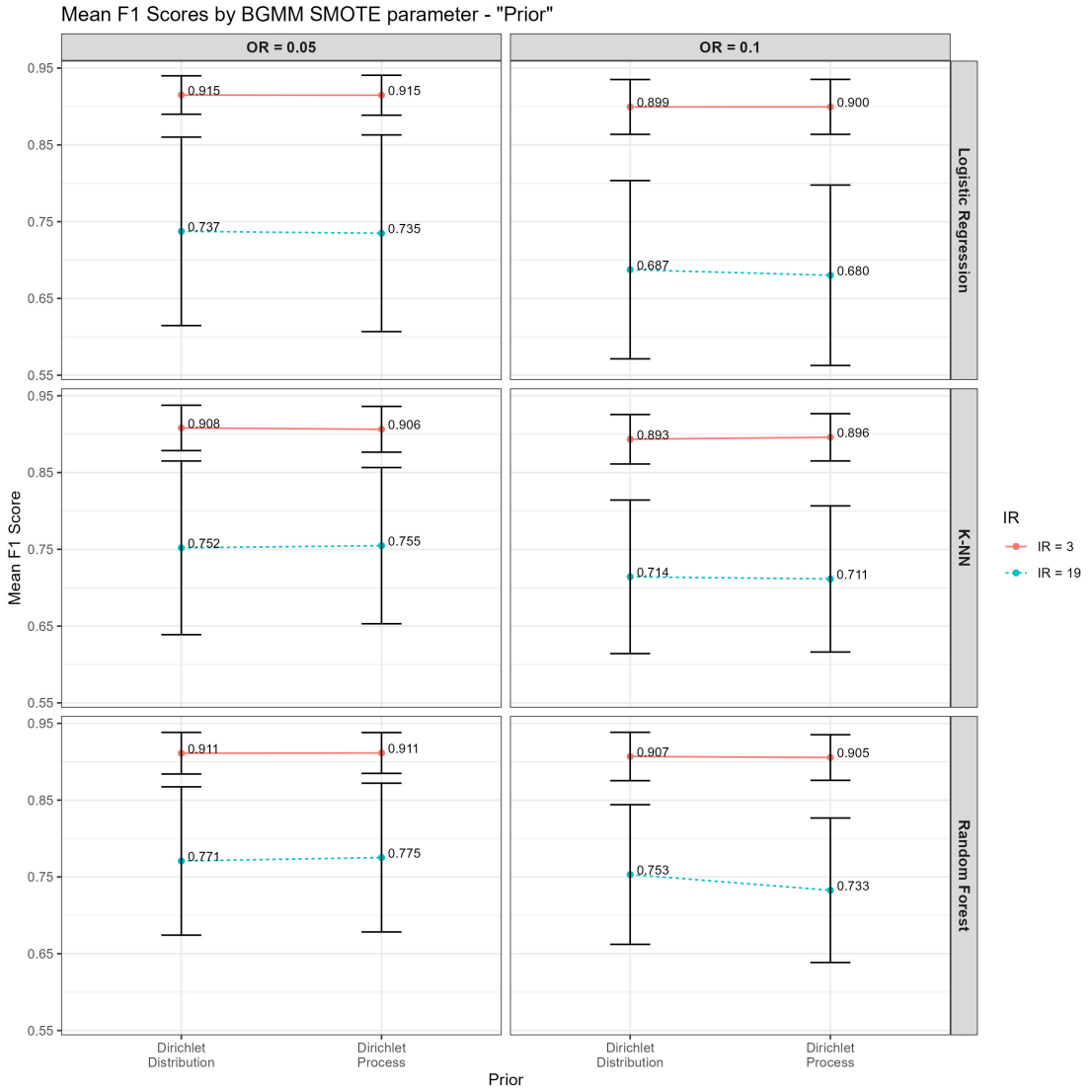


Figure A.10: Mean F1 Scores for different priors of BGMM SMOTE for two different outlier ratios and two different imbalance ratios.

A.4 PCA Plots of Outliers Detected by LOF

The Fig. A.11, Fig. A.12, Fig. A.13, and Fig. A.14 show the Principal Component Analysis (PCA) plots for the first two components, highlighting the distribution of outliers within the minority class across 25 real-world datasets analyzed in the experimental study. The outliers were detected by the Local Outlier Factor (LOF) method for each application dataset used in the analysis.

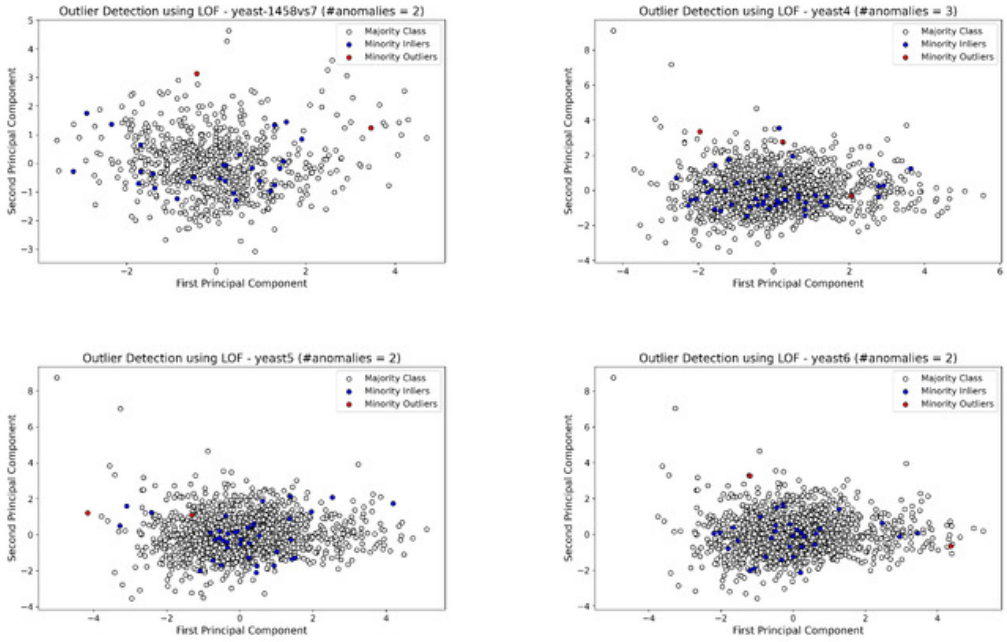


Figure A.11: Principal component plots for the first two components, showing the distribution of outliers within the minority class across 4 datasets detected by the LOF method.

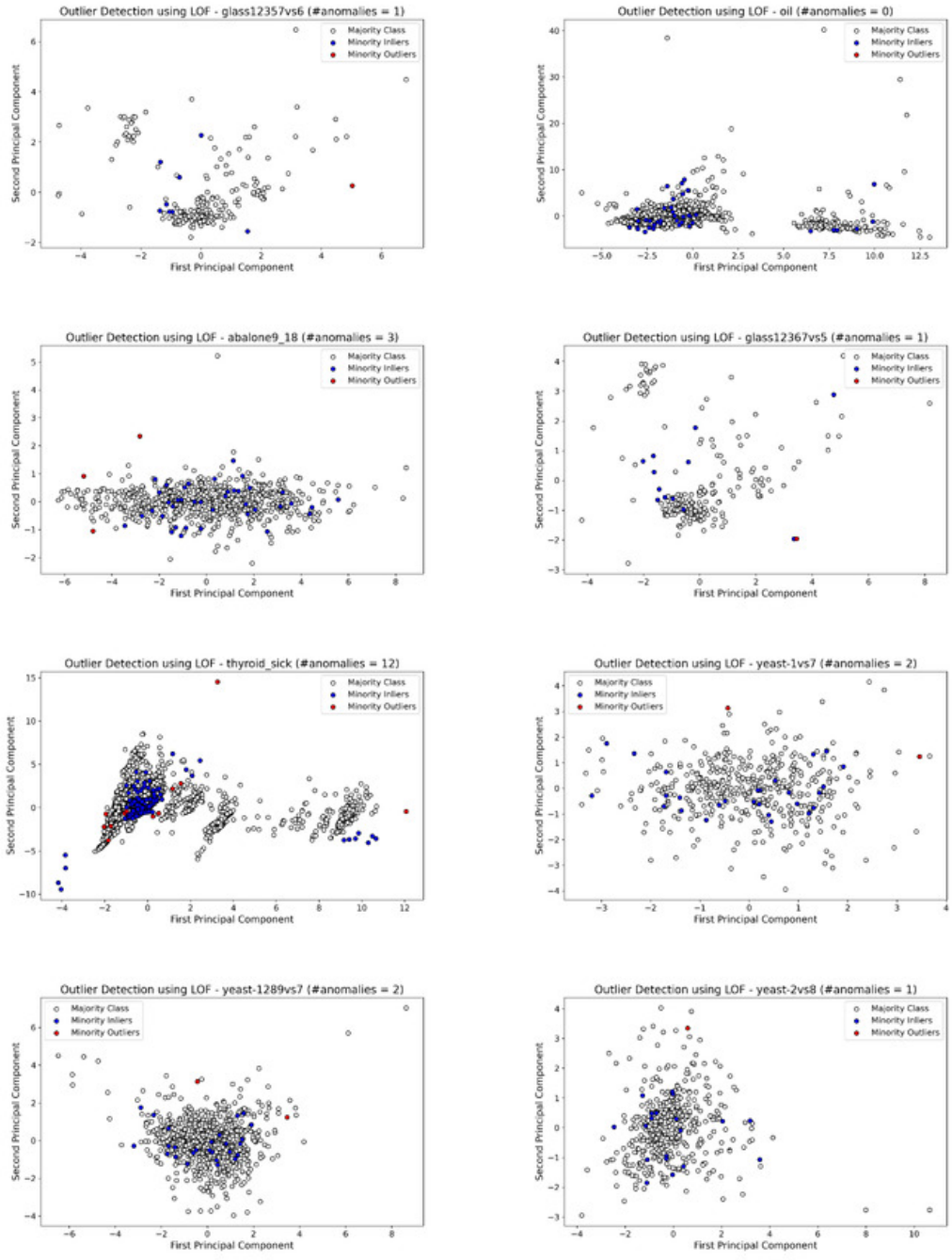


Figure A.12: Principal component plots for the first two components, showing the distribution of outliers within the minority class across 8 datasets detected by the LOF method.

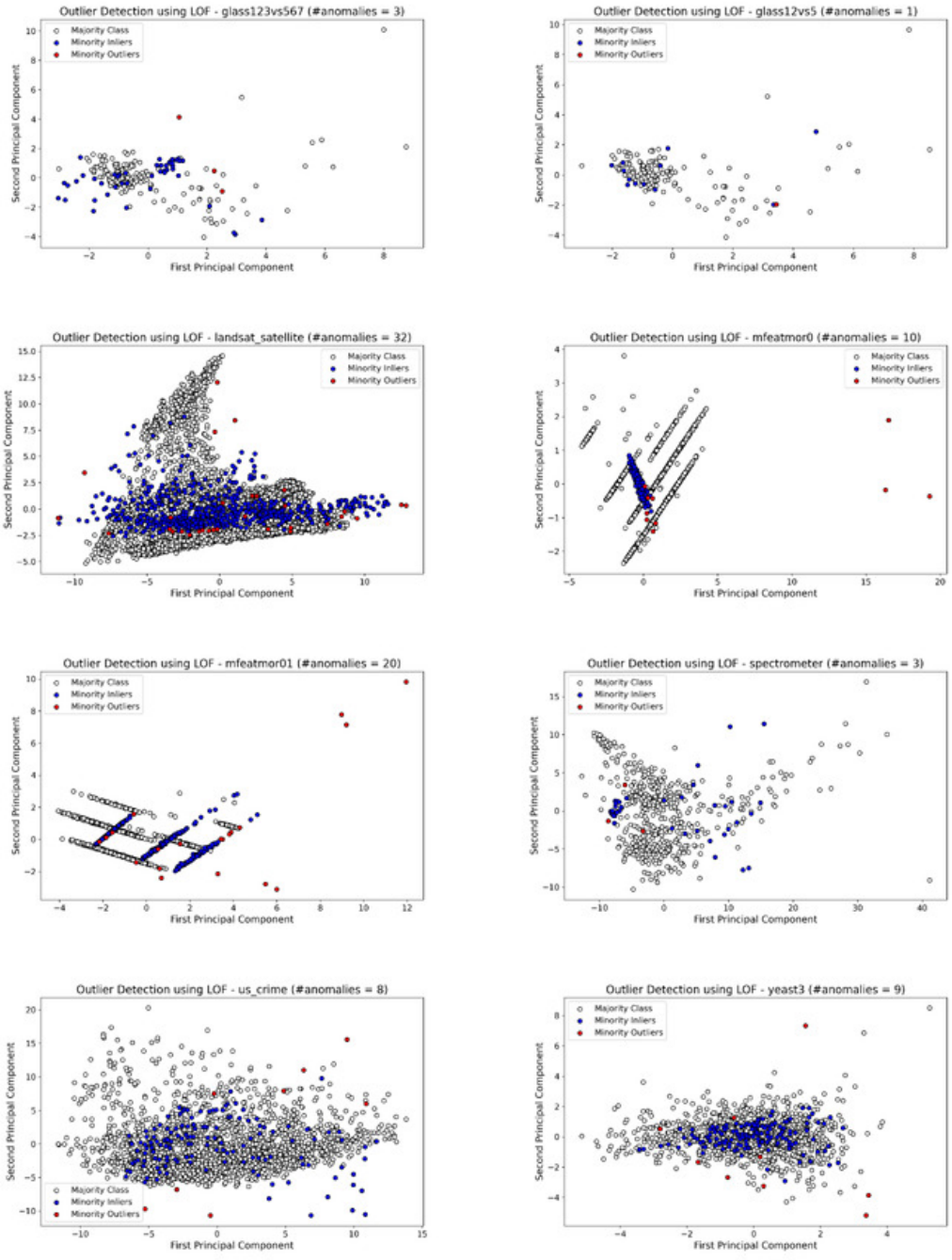


Figure A.13: Principal component plots for the first two components, showing the distribution of outliers within the minority class across 8 datasets detected by the LOF method.

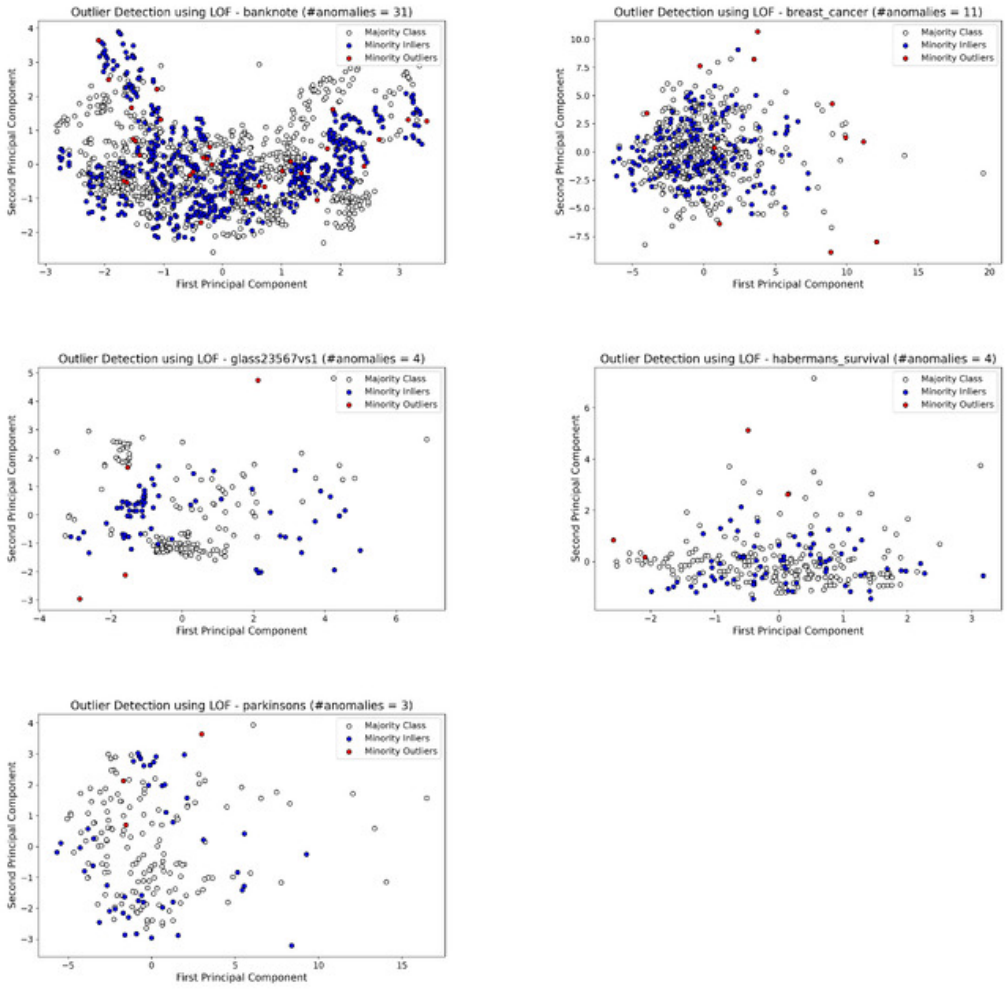


Figure A.14: Principal component plots for the first two components, showing the distribution of outliers within the minority class across 6 datasets detected by the LOF method.

A.5 Comprehensive Evaluation with Additional Accuracy Measures

In addition to evaluating our methods based solely on the F1 score, we conducted a comprehensive assessment using a variety of performance metrics to ensure a robust evaluation of our classifiers. These metrics included recall, precision, PR AUC (Precision-Recall Area Under Curve), and MCC (Matthews Correlation Coefficient). By incorporating these additional measures, we aimed to capture different aspects of classifier performance, providing a more holistic understanding of their effectiveness.

Table A.1, Table A.2, Table A.3, Table A.4, Table A.5, and Table A.6 present the detailed results for PR AUC and MCC for each of the three classifiers: Logistic Regression, k-Nearest Neighbors (k-NN), and Random Forest. These tables illustrate how each classifier performs under different conditions, allowing for a nuanced comparison of their capabilities. By analyzing these metrics, we demonstrate the strengths and limitations of each classifier and the effectiveness of our methods across various scenarios.

A.5.1 MCC

Table A.1: MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Logistic Regression classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.4204	0.4229	**0.4547**	0.4075	0.3637	0.3646	0.3646	0.4434	0.3653	0.4379	0.3582	0.3265	0.3079	0.3294	0.4417	0.3248	0.4434
yeast5	0.4503	0.4462	0.465	0.4354	0.4161	0.4137	0.4124	0.5051	0.4112	**0.5508**	0.3978	0.4441	0.3688	0.379	0.4663	0.3652	0.5051
yeast-1289vs7	0.2013	0.2443	0.2301	0.1977	0.2054	0.2053	0.1965	0.2474	0.1949	0.0745	0.1939	0.206	**0.2522**	0.2329	0.2398	0.1955	0.2474
yeast4	0.2727	0.3136	0.3269	0.2927	0.2929	0.2948	0.2947	0.299	0.2976	**0.3846**	0.2987	0.2482	0.2903	0.3101	0.3618	0.2746	0.299
yeast-2vs8	0.2375	0.3891	0.2769	0.2636	0.4682	0.4669	0.4455	0.6263	0.4433	0.2867	0.4961	**0.6363**	0.6082	0.4954	0.2087	0.1447	0.6263
glass12357vs6	0.5223	0.6464	0.5718	0.5924	0.6321	0.6321	0.6321	0.4875	0.6321	**0.6806**	0.6321	0.4346	0.6268	0.4896	0.6321	0.6321	0.4875
yeast-1458vs7	0.1446	**0.1982**	0.1516	0.1732	0.1661	0.1637	0.1602	0	0.1576	0.0904	0.1302	0.1456	0.1233	0.1532	0.173	0.1573	0
oil	0.5422	0.5416	0.4974	0.5027	0.4975	0.4978	0.5033	**0.5534**	0.5023	0.5102	0.5132	0.4444	0.4897	0.415	0.4642	0.4966	**0.5534**
abalone9_18	0.4987	0.5005	0.4748	0.4798	0.4394	0.4417	0.4549	0.4929	0.4549	0.5189	0.4838	0.5067	0.471	0.364	**0.5277**	0.4108	0.4929
glass12367vs5	0.4423	0.5437	0.5447	0.4806	0.5267	0.5259	0.519	0.3422	0.5208	0.1989	0.5458	0.3158	0.408	0.4879	**0.563**	0.5375	0.3422
thyroid_sick	0.5797	0.5861	0.5583	0.5715	0.5571	0.5583	0.5548	**0.6935**	0.5542	0.6927	0.5838	0.5754	0.6225	0.3766	0.5373	0.5298	**0.6935**
yeast-1vs7	0.3385	0.3313	**0.3586**	0.3194	0.2887	0.2897	0.2928	0.1763	0.2941	0.2171	0.287	0.3135	0.3137	0.3303	0.3051	0.293	0.1763
us_crime	0.4564	0.4673	0.4689	0.47	0.4623	0.4607	0.4571	0.4744	0.4589	0.4615	0.4778	0.4972	0.4757	0.4706	**0.4988**	0.448	0.4744
glass12vs5	0.6241	0.785	0.687	0.7652	0.776	0.7743	0.7706	0.7162	0.7705	0.4701	0.7874	0.6046	0.7459	0.654	0.7329	**0.791**	0.7162
spectrometer	0.8451	0.8479	0.8513	0.846	0.8361	0.8336	0.8381	0.8534	0.8379	0.8375	0.8467	0.8284	0.8485	0.7818	0.8394	0.8209	**0.8607**
landsat_satellite	0.2371	0.245	0.2467	0.2485	**0.2661**	0.2654	0.2646	0.0945	0.2655	0.0592	0.261	0.0656	0.2568	0.257	0.2604	0.2579	0.0945
mfeatmor0	0.9916	0.9924	0.9916	0.9916	0.9908	0.9915	0.9901	0.99	0.9905	0.9916	**0.9939**	0.9709	0.9858	0.9741	0.9916	0.9889	0.9916
yeast3	0.6754	0.6713	0.6757	0.6658	0.6231	0.6244	0.6251	**0.7293**	0.6251	0.6789	0.6411	0.7031	0.6225	0.6304	0.6244	0.6183	**0.7293**
mfeatmor01	0.8966	0.8969	0.8917	0.8947	0.8853	0.8898	0.89	0.9053	0.8854	0.9049	0.8844	0.8855	0.8833	0.8291	0.797	0.7088	**0.9084**
glass123vs567	0.7872	0.8166	**0.8267**	0.8171	0.7965	0.7948	0.7973	0.737	0.7982	0.7993	0.7829	0.7978	0.7961	0.7807	0.7842	0.7892	0.7329
parkinsons	0.5496	0.5747	0.5587	0.557	0.5444	0.5469	0.5431	**0.6491**	0.5462	0.5895	0.5266	0.6076	0.5427	0.4987	0.5567	0.5584	**0.6491**
habermans_survival	0.262	0.3154	0.2858	0.2873	0.3012	0.304	0.301	0.2233	0.3061	0.2756	0.3109	0.145	**0.3241**	0.2923	0.2791	0.3111	0.2233
glass23567vs1	0.4348	0.454	0.4398	0.435	0.4203	0.4217	0.4267	**0.4957**	0.4179	0.4172	0.4348	0.4879	0.4496	0.4546	0.4507	0.4458	**0.4957**
breast_cancer	0.9462	0.9485	0.9493	0.9467	0.9438	0.9435	0.9422	**0.9496**	0.9444	0.9471	0.9408	0.8996	0.9431	0.9431	0.9223	0.9211	0.9484
banknote	0.9799	**0.9826**	0.9825	0.9815	0.9809	0.9808	0.981	0.981	0.9808	0.9805	0.9812	0.903	0.9811	0.95	0.981	NA	0.981

Table A.2: MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a K-NN classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.4748	0.506	**0.5154**	0.4983	0.4	0.4	0.3952	0.499	0.3979	0.442	0.3741	0.4653	0.4022	0.3962	0.3947	0.3571	0.499
yeast5	0.512	0.5941	0.549	0.57	0.4954	0.4924	0.4868	**0.6553**	0.4905	0.6444	0.4487	0.6018	0.5092	0.4226	0.5322	0.4464	**0.6553**
yeast-1289vs7	0.111	0.2443	0.2216	0.2048	0.2224	0.2279	0.2225	0.2337	0.2251	0.11	0.2074	**0.3196**	0.2506	0.221	0.2172	0.2207	0.2337
yeast4	0.3298	**0.3957**	0.3675	0.3705	0.334	0.3328	0.3455	0.3066	0.3398	0.3087	0.3304	0.363	0.3296	0.3689	0.3431	0.3272	0.3066
yeast-2vs8	0.5133	0.5437	0.4461	0.4333	0.4019	0.4019	0.3746	**0.6461**	0.3799	0.6009	0.4167	0.6287	0.4332	0.4122	0.3266	0.3644	**0.6461**
glass12357vs6	**0.8174**	0.7875	0.6857	0.7665	0.7864	0.7897	0.7945	0.7571	0.7955	0.7571	0.7352	0.6961	0.7462	0.5423	0.7862	0.7859	0.7571
yeast-1458vs7	0.0044	**0.1616**	0.0069	0.0444	0.0982	0.0985	0.1007	0.0162	0.1055	0.1257	0.1277	0.1354	0.1132	0.0761	0.1024	0.0899	0.0162
oil	0.492	0.5941	0.551	0.5492	0.5559	0.5586	0.5583	0.5435	0.5594	0.5917	0.5561	0.3559	0.5431	0.4199	**0.6071**	0.5478	0.5435
abalone9_18	0.3351	0.4082	**0.4286**	0.4273	0.3791	0.378	0.3825	0.4267	0.3831	0.4058	0.3823	0.4239	0.3404	0.3716	0.4025	0.3533	0.4267
glass12367vs5	0.6008	0.6697	0.6234	0.6628	0.6548	0.6624	0.6739	0.4325	**0.6836**	0.2077	0.6465	0.3754	0.562	0.5671	0.6442	0.6261	0.4325
thyroid_sick	0.5266	0.5422	0.5458	0.5464	0.5561	0.5537	0.5514	0.5378	0.5554	0.4877	0.5192	0.487	0.5458	0.489	0.538	**0.5566**	0.5378
yeast-1vs7	0.303	0.392	0.3067	0.3512	0.3773	0.3753	0.3652	**0.4295**	0.3663	0.2494	0.3277	0.3994	0.4096	0.3211	0.4094	0.3529	**0.4295**
us_crime	0.3367	0.3667	0.3649	0.3711	0.3583	0.3579	0.3613	0.3951	0.3565	0.3051	0.3941	**0.3997**	0.3675	0.3673	0.3607	0.3453	0.3951
glass12vs5	0.2675	0.5669	0.5185	0.5487	0.6376	0.6218	0.6553	0.293	0.6513	0.2017	0.6474	0.2765	0.6	**0.6847**	0.6416	0.6435	0.293
spectrometer	0.7894	**0.8678**	0.7576	0.8235	0.8332	0.8384	0.8401	0.8273	0.8374	0.8071	0.8135	0.7958	0.8185	0.8667	0.789	0.8134	0.8388
landsat_satellite	0.6246	**0.6736**	0.6314	0.6549	0.6702	0.6702	0.6663	0.6723	0.6674	0.6703	0.6531	0.4775	0.6614	0.5948	0.6559	0.6667	0.6723
mfeatmor0	0.9916	**0.9926**	0.9916	0.9916	0.9832	0.9877	0.9846	0.9889	0.9855	0.9916	0.9874	0.9462	0.9864	0.9611	0.9916	0.9862	0.9916
yeast3	0.6914	0.7123	0.7216	0.7161	0.685	0.685	0.6904	**0.7322**	0.688	0.66	0.7308	0.7191	0.6572	0.6851	0.5983	0.6508	**0.7322**
mfeatmor01	0.9367	0.9383	0.9389	0.9352	0.9321	0.9319	0.9332	0.939	0.933	0.9222	0.932	0.9223	0.9277	0.928	0.8995	0.8603	**0.9453**
glass123vs567	0.8258	**0.84**	0.8355	0.8363	0.8118	0.8103	0.8139	0.8165	0.8099	0.8095	0.8331	0.7882	0.8045	0.8329	0.8017	0.8024	0.8189
parkinsons	0.8288	**0.8291**	0.8071	0.821	0.7827	0.7847	0.7825	0.7039	0.7897	0.6765	0.7437	0.6935	0.79	0.7914	0.8068	0.8255	0.7039
habermans_survival	0.0453	0.1874	0.1529	0.149	0.1839	0.1827	0.1704	-0.003	0.1854	0.0551	0.2153	0.1122	0.1753	**0.2307**	0.1848	0.178	-0.003
glass23567vs1	0.603	**0.6387**	0.6167	0.6148	0.5771	0.5764	0.5776	0.5419	0.5755	0.5309	0.5351	0.531	0.6139	0.5251	0.5875	0.5954	0.5419
breast_cancer	0.9175	**0.9286**	0.9211	0.9202	0.9174	0.9186	0.9145	0.9128	0.9186	0.9112	0.9146	0.8842	0.9149	0.9124	0.9038	0.9042	0.9125
banknote	0.9971	**0.9976**	0.9971	0.9971	0.9971	0.9971	0.9971	0.9971	0.9971	0.9971	0.9967	0.9126	0.9971	0.9747	0.9971	NA	0.9971

Table A.3: MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5556	0.6105	**0.63**	0.6039	0.5424	0.5487	0.5426	0.5173	0.5485	0.5212	0.4931	0.5172	0.5608	0.5364	0.5231	0.5281	0.5157
yeast5	0.6115	0.6269	0.6228	0.6273	0.5235	0.5273	0.5194	0.5986	0.5144	0.5904	0.5131	**0.6549**	0.5472	0.6157	0.5408	0.5068	0.5988
yeast-1289vs7	0.0584	0.1397	0.0719	0.097	0.1534	0.1504	0.1599	0.2947	0.1505	0.1165	0.1292	0.3183	0.2688	**0.3463**	0.2162	0.119	0.2924
yeast4	0.2814	0.3899	0.3638	0.366	0.392	0.3896	0.3779	0.3551	0.3729	0.3637	0.3808	0.3413	0.4245	0.3983	**0.4299**	0.3814	0.359
yeast-2vs8	0.5919	0.6001	0.5337	0.4813	0.4381	0.4372	0.4298	0.6131	0.429	0.6306	0.5114	0.6227	0.5445	**0.659**	0.3427	0.2806	0.6139
glass12357vs6	0.7149	**0.7347**	0.6629	0.6485	0.6165	0.6222	0.616	0.5771	0.609	0.6192	0.6369	0.5408	0.6209	0.6143	0.6219	0.5968	0.5826
yeast-1458vs7	-0.0037	0.1539	0.0094	0.0573	0.0987	0.0924	0.0924	-0.0032	0.0841	-0.0117	0.0937	**0.1614**	0.0727	0.0327	0.1103	0.0889	0.0005
oil	0.4578	0.5413	0.4939	0.5658	0.5014	0.5066	0.5128	0.4817	0.5136	0.3278	**0.5916**	0.4637	0.4484	0.5656	0.5372	0.4928	0.4819
abalone9_18	0.3663	0.3945	0.3683	0.3724	0.3395	0.3387	0.3454	0.3763	0.3436	**0.4026**	0.3681	0.3966	0.3452	0.3482	0.3739	0.3219	0.3821
glass12367vs5	0.659	**0.8418**	0.6662	0.7232	0.7437	0.7417	0.7552	0.6031	0.7555	0.7286	0.7788	0.6221	0.682	0.7265	0.7619	0.7412	0.6024
thyroid_sick	0.8521	**0.8769**	0.8625	0.8515	0.8673	0.8626	0.8646	0.853	0.8652	0.85	0.8402	0.8061	0.871	0.8631	0.8609	0.8633	0.8521
yeast-1vs7	0.1528	0.3019	0.1863	0.2536	0.322	0.3199	0.303	0.3373	0.3171	0.1233	0.2151	**0.3931**	0.3295	0.3575	0.3265	0.2724	0.3373
us_crime	0.4449	0.4574	0.4364	0.4577	0.462	0.4618	0.4588	0.4768	0.4621	0.3822	0.4815	0.4711	0.4538	**0.4851**	0.4611	0.451	0.4799
glass12vs5	0.671	**0.8069**	0.7643	0.7802	0.7811	0.7884	0.7845	0.692	0.7824	0.7485	0.7843	0.5826	0.7265	0.7773	0.7821	0.7786	0.6922
spectrometer	0.8188	**0.8466**	0.798	0.8201	0.8003	0.8094	0.8018	0.8314	0.8002	0.8306	0.7939	0.8062	0.8013	0.8102	0.8011	0.7839	0.825
landsat_satellite	0.6254	0.6456	0.6401	0.6394	0.6529	0.6483	0.6503	0.6454	0.6505	0.6441	0.6456	0.6262	**0.6572**	0.5997	0.6511	0.6523	0.6459
mfeatmor0	0.9916	**0.9918**	0.9915	0.9916	0.9878	0.9905	0.9875	0.988	0.9871	0.9901	0.9903	0.9498	0.9864	0.9867	0.9916	0.9858	0.9916
yeast3	0.7402	**0.7618**	0.7522	0.757	0.7584	0.7599	0.7592	0.7339	0.7599	0.7429	0.7617	0.731	0.7574	0.7424	0.7516	0.7496	0.7352
mfeatmor01	0.9413	0.9477	0.9453	0.9406	0.9384	0.9393	0.938	0.9419	0.9379	0.9436	0.9463	0.9241	0.9358	0.9426	0.9154	0.8663	**0.9481**
glass123vs567	0.86	**0.8762**	0.8598	0.8707	0.856	0.8559	0.854	0.8548	0.8627	0.8559	0.8659	0.7772	0.8549	0.8586	0.8415	0.8387	0.8542
parkinsons	0.7022	0.7584	0.7391	0.7437	0.7538	0.7564	0.7564	0.7372	0.7576	0.7434	0.7239	0.669	**0.7608**	0.7277	0.7427	0.7585	0.7369
habermans_survival	0.1934	0.2194	0.1988	0.2272	0.1767	0.1728	0.1933	0.137	0.1928	0.1333	0.2234	0.1244	0.181	**0.2525**	0.1649	0.1698	0.1354
glass23567vs1	0.7274	**0.7765**	0.747	0.7408	0.7343	0.7349	0.7307	0.7372	0.7284	0.75	0.6901	0.736	0.7322	0.6838	0.7263	0.7228	0.7437
breast_cancer	0.9172	**0.9246**	0.9183	0.92	0.9163	0.9166	0.9186	0.9132	0.9166	0.9169	0.9148	0.8856	0.917	0.9147	0.9115	0.9206	0.9136
banknote	0.9873	**0.9891**	0.9873	0.9883	0.9867	0.9864	0.9866	0.986	0.9868	0.9864	0.9864	0.8965	0.9867	0.9822	0.986	NA	0.9861

A.5.2 PR-AUC

Table A.4: PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100×5 fold cross-validation with a Logistic Regression classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5159	0.5168	0.5392	0.5122	0.4988	0.4987	0.5027	**0.5758**	0.5039	0.5056	0.5092	0.3705	0.5016	0.5076	0.5129	0.4936	**0.5758**
yeast5	0.5521	0.5454	0.5507	0.5384	0.5373	0.5373	0.536	0.5626	0.5343	**0.5789**	0.5425	0.4867	0.4964	0.5347	0.5456	0.523	0.5626
yeast-1289vs7	0.3738	0.3786	0.4231	0.3798	0.406	0.407	0.4043	**0.4304**	0.4036	0.2161	0.4043	0.251	0.4188	0.4151	0.3916	0.4111	**0.4304**
yeast4	0.4274	0.4452	0.4407	0.4307	0.4477	0.4469	0.4452	0.3931	0.4485	0.4396	0.4598	0.2966	0.4327	**0.4851**	0.473	0.4364	0.3931
yeast-2vs8	0.4275	0.4863	0.4661	0.4439	0.5404	0.5417	0.5248	**0.7204**	0.525	0.4547	0.5573	0.6943	0.6402	0.5531	0.4242	0.4311	**0.7204**
glass12357vs6	0.696	**0.7865**	0.732	0.7472	0.7797	0.7797	0.7797	0.632	0.7797	0.7499	0.7797	0.5878	0.7752	0.6396	0.7797	0.7797	0.632
yeast-1458vs7	0.3133	0.3817	0.3218	0.3543	0.3794	0.3745	0.379	**0.5216**	0.3784	0.2888	0.3687	0.2279	0.3626	0.3081	0.3559	0.374	**0.5216**
oil	0.5758	0.5766	0.5395	0.5439	0.565	0.5652	0.5698	**0.5931**	0.5687	0.5798	0.5816	0.4934	0.5585	0.5317	0.5112	0.5646	**0.5931**
abalone9_18	0.5728	0.5782	0.5687	0.5679	0.5662	0.5665	0.5726	0.5728	0.5727	0.5662	0.5833	0.5971	0.5812	0.4768	**0.6093**	0.556	0.5728
glass12367vs5	0.4898	0.5948	0.5927	0.5296	0.5915	0.5902	0.5837	0.402	0.5849	0.3566	0.612	0.3815	0.48	0.5983	**0.617**	0.6031	0.402
thyroid_sick	0.6374	0.6412	0.6272	0.634	0.6358	0.6325	0.6345	**0.725**	0.6344	0.7214	0.653	0.6204	0.6704	0.5581	0.6267	0.6276	**0.725**
yeast-1vs7	0.4576	0.4577	**0.4868**	0.4493	0.4667	0.468	0.4681	0.345	0.4697	0.329	0.4607	0.4357	0.4574	0.4794	0.4394	0.4806	0.345
us_crime	0.535	0.5477	0.5472	0.5499	0.5601	0.5579	0.555	0.5428	0.5567	0.5663	0.5697	0.561	0.5618	**0.5893**	0.5704	0.5537	0.5428
glass12vs5	0.6741	0.8139	0.7321	0.7969	0.8054	0.8038	0.8002	0.7511	0.8001	0.5316	0.8174	0.661	0.7839	0.724	0.7677	**0.82**	0.7511
spectrometer	0.8691	0.871	0.872	0.8697	0.8593	0.8575	0.861	0.8764	0.8607	0.8623	0.8679	0.8624	0.8696	0.8159	0.8615	0.846	**0.8835**
landsat_satellite	0.4919	0.5039	0.5098	0.5144	0.5463	0.5467	0.5447	0.3316	0.5455	0.201	0.5281	0.198	0.5225	**0.5722**	0.5374	0.5408	0.3316
mfeatmor0	0.9932	0.9939	0.9932	0.9932	0.9922	0.9929	0.9916	0.9918	0.9919	0.9932	**0.995**	0.9767	0.9878	0.9777	0.9932	0.9906	0.9932
yeast3	0.7372	0.7343	0.7379	0.7306	0.712	0.7131	0.7132	**0.7729**	0.7131	0.7372	0.7141	0.7537	0.7085	0.7174	0.7157	0.7143	**0.7729**
mfeatmor01	0.9237	0.924	0.92	0.9224	0.9159	0.9191	0.9192	0.9316	0.9159	0.9305	0.9157	0.9215	0.9135	0.8753	0.8546	0.7935	**0.9361**
glass123vs567	0.8608	0.877	**0.8847**	0.8781	0.8625	0.8612	0.8629	0.829	0.8635	0.8682	0.8529	0.8719	0.8621	0.8517	0.8532	0.8565	0.8265
parkinsons	0.7053	0.7281	0.7111	0.7124	0.7103	0.7117	0.7088	**0.795**	0.7112	0.7295	0.7003	0.7707	0.7108	0.6817	0.719	0.7228	**0.795**
habermans_survival	0.5785	**0.5973**	0.5882	0.5844	0.5699	0.5721	0.5713	0.4906	0.5726	0.5649	0.571	0.4425	0.5823	0.5519	0.5612	0.583	0.4906
glass23567vs1	0.6938	0.7081	0.7053	0.6997	0.69	0.6912	0.6924	**0.7168**	0.6891	0.6884	0.6935	0.7115	0.7005	0.6998	0.7108	0.7105	**0.7168**
breast_cancer	0.9764	0.9772	0.9783	0.9766	0.9748	0.9747	0.9741	0.9782	0.9751	0.9776	0.9728	0.9565	0.9746	0.9727	0.9601	0.9585	**0.9785**
banknote	0.9907	0.9923	0.9933	0.9918	0.9937	0.9936	0.9937	**0.9941**	0.9936	0.9935	0.9935	0.9675	0.9938	0.9764	**0.9941**	NA	**0.9941**

Table A.5: PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100×5 fold cross-validation with a K-NN classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5039	0.5293	**0.5404**	0.5235	0.4336	0.4336	0.4301	0.526	0.4319	0.4658	0.4363	0.499	0.4366	0.4871	0.4244	0.396	0.526
yeast5	0.5413	0.614	0.5726	0.595	0.5339	0.5318	0.5282	**0.6832**	0.5304	0.6702	0.518	0.6309	0.5404	0.5096	0.557	0.4986	**0.6832**
yeast-1289vs7	0.1504	0.2851	0.2661	0.2706	0.2844	0.2896	0.2896	0.4304	0.2921	0.1815	0.3623	**0.4317**	0.2999	0.2992	0.263	0.285	0.4304
yeast4	0.381	0.4329	0.4089	0.4083	0.3732	0.3724	0.386	0.3728	0.3807	0.3596	0.4296	0.4267	0.366	**0.438**	0.3795	0.3684	0.3728
yeast-2vs8	0.5964	0.5803	0.5145	0.5066	0.4722	0.473	0.4452	**0.726**	0.453	0.6604	0.496	0.7193	0.4792	0.4881	0.3818	0.4328	**0.726**
glass12357vs6	**0.8547**	0.8181	0.7446	0.8073	0.8103	0.8137	0.8175	0.807	0.8185	0.8047	0.7708	0.8071	0.7822	0.6488	0.8103	0.8092	0.807
yeast-1458vs7	0.0543	0.2236	0.072	0.1114	0.165	0.1663	0.1714	0.1626	0.1774	0.3035	0.2638	**0.3803**	0.1763	0.1703	0.1606	0.1577	0.1626
oil	0.5408	0.63	0.5915	0.5815	0.5886	0.5908	0.5909	0.5833	0.5921	0.6324	0.5919	0.4367	0.5805	0.4946	**0.6395**	0.582	0.5833
abalone9_18	0.397	0.4714	0.4815	0.492	0.4502	0.4502	0.454	**0.5028**	0.4563	0.4858	0.4685	0.4912	0.4054	0.4372	0.4675	0.4254	**0.5028**
glass12367vs5	0.683	0.715	0.6674	0.7034	0.7042	**0.7271**	0.715	0.522	0.7183	0.559	0.6779	0.5861	0.6926	0.6525	0.7104	0.6589	0.522
thyroid_sick	0.5893	0.591	0.5941	0.5925	0.5984	0.5975	0.5944	0.5946	**0.5985**	0.583	0.5644	0.5491	0.5902	0.5738	0.582	0.5983	0.5946
yeast-1vs7	0.3686	0.4652	0.376	0.427	0.4576	0.4543	0.454	0.4896	0.4578	0.4211	0.449	**0.5223**	0.4791	0.4136	0.4824	0.428	0.4896
us_crime	0.4205	0.4505	0.4434	0.4513	0.4314	0.431	0.4357	**0.4939**	0.4315	0.3813	0.4605	0.4819	0.4379	0.439	0.4322	0.4216	**0.4939**
glass12vs5	0.448	0.6247	0.5843	0.6096	0.6885	0.6734	0.704	0.3668	0.7005	0.4678	0.697	0.4901	0.6559	**0.7454**	0.7015	0.6936	0.3668
spectrometer	0.8196	**0.8879**	0.7993	0.8499	0.8584	0.8631	0.8634	0.8543	0.8608	0.8423	0.8398	0.8287	0.8486	0.8854	0.8263	0.8425	0.8635
landsat_satellite	0.6895	0.7197	0.695	0.712	0.7202	0.7203	0.7179	0.7189	0.7187	0.7171	0.7045	0.5535	0.714	0.6931	0.71	**0.7213**	0.7189
mfeatmor0	0.9932	**0.9941**	0.9932	0.9932	0.9857	0.9898	0.987	0.9908	0.9878	0.9932	0.9895	0.9572	0.9886	0.9661	0.9932	0.9885	0.9932
yeast3	0.7475	0.7633	0.7685	0.7645	0.7396	0.7393	0.7441	**0.7776**	0.7419	0.7263	0.7752	0.7685	0.7184	0.7415	0.6655	0.7076	**0.7776**
mfeatmor01	0.9555	0.9564	0.9573	0.9534	0.9508	0.9506	0.9516	0.9559	0.9514	0.9439	0.9501	0.9482	0.9474	0.9469	0.9264	0.8968	**0.9613**
glass123vs567	0.8877	**0.8949**	0.8936	0.8932	0.8751	0.8744	0.877	0.8783	0.8744	0.873	0.8874	0.8642	0.8703	0.8861	0.8718	0.8708	0.88
parkinsons	**0.8967**	0.8962	0.8793	0.8884	0.8625	0.8634	0.8624	0.8116	0.8667	0.7986	0.8323	0.8139	0.8681	0.8608	0.8765	0.8883	0.8116
habermans_survival	0.3881	0.4952	0.4722	0.4706	0.4947	0.4945	0.4899	0.3079	0.4958	0.3867	0.5228	0.4188	0.4755	**0.5378**	0.4953	0.4993	0.3079
glass23567vs1	0.7802	**0.799**	0.7864	0.7851	0.7643	0.7641	0.7649	0.7477	0.7637	0.7409	0.7425	0.739	0.7865	0.7394	0.7709	0.7745	0.7477
breast_cancer	0.9597	**0.9649**	0.961	0.961	0.9585	0.9591	0.9567	0.956	0.9591	0.9539	0.9568	0.9491	0.9576	0.9545	0.9527	0.9523	0.9561
banknote	0.9984	**0.9987**	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9982	0.9709	0.9984	0.9862	0.9984	NA	0.9984

Table A.6: PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100×5 fold cross-validation with a Random Forest classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5985	0.6371	**0.6574**	0.6272	0.5672	0.5727	0.5671	0.5509	0.572	0.5547	0.524	0.5489	0.5812	0.5589	0.5464	0.5544	0.5486
yeast5	0.6451	0.6567	0.6519	0.6536	0.5511	0.5539	0.5479	0.6262	0.543	0.6217	0.5529	**0.6884**	0.5738	0.636	0.5865	0.5361	0.6254
yeast-1289vs7	0.2089	0.1979	0.1151	0.1416	0.1963	0.1934	0.2029	0.3847	0.1935	0.3645	0.2027	0.3992	0.3096	**0.408**	0.2573	0.1633	0.3803
yeast4	0.3972	0.4346	0.4023	0.4019	0.4275	0.4246	0.4139	0.4329	0.4094	0.443	0.4489	0.4008	0.4639	0.4465	**0.4701**	0.4173	0.4364
yeast-2vs8	0.6874	0.6841	0.6127	0.5519	0.5006	0.5016	0.4952	0.7086	0.492	0.7182	0.5856	0.7161	0.6208	**0.7374**	0.4123	0.3399	0.7098
glass12357vs6	0.7966	**0.8043**	0.7837	0.767	0.7786	0.7764	0.774	0.744	0.7742	0.7533	0.7903	0.7324	0.7865	0.6644	0.7834	0.7579	0.7459
yeast-1458vs7	0.3463	0.2426	0.0645	0.1196	0.1564	0.1509	0.1521	0.3748	0.1439	0.1689	0.1794	**0.3982**	0.1256	0.2082	0.1653	0.1471	0.3806
oil	0.5564	0.6061	0.5507	0.6258	0.5707	0.5748	0.5795	0.5672	0.5807	0.5084	**0.6505**	0.5644	0.5265	0.6207	0.5882	0.5718	0.5615
abalone9_18	0.4409	0.4563	0.4393	0.4374	0.4048	0.4039	0.4099	0.4582	0.4083	0.4755	0.4471	**0.4875**	0.4204	0.4243	0.4375	0.3885	0.4638
glass12367vs5	0.7892	**0.8722**	0.7957	0.8054	0.8031	0.804	0.8141	0.7448	0.8131	0.797	0.8284	0.7601	0.7729	0.7668	0.8156	0.7997	0.7461
thyroid_sick	0.8688	**0.8892**	0.8773	0.8669	0.8799	0.8758	0.8775	0.8692	0.878	0.8678	0.8557	0.8284	0.8839	0.8778	0.8748	0.8762	0.8684
yeast-1vs7	0.3842	0.4086	0.2619	0.3309	0.3883	0.3881	0.3724	0.4836	0.3853	0.3734	0.3121	**0.5329**	0.4009	0.4374	0.3938	0.3446	0.4886
us_crime	0.5185	0.5242	0.5059	0.521	0.5231	0.5227	0.5203	0.5476	0.5232	0.506	0.5421	0.5432	0.5162	0.5441	0.5231	0.513	**0.55**
glass12vs5	0.7239	**0.845**	0.8024	0.8171	0.8193	0.8254	0.8211	0.7539	0.8198	0.7976	0.821	0.7089	0.7718	0.8134	0.8199	0.8173	0.7556
spectrometer	0.8507	**0.8742**	0.8301	0.8482	0.8284	0.8375	0.8292	0.8603	0.828	0.8639	0.8219	0.845	0.8307	0.8381	0.8306	0.8144	0.8601
landsat_satellite	0.6831	0.698	0.6932	0.6916	0.7035	0.6994	0.7009	0.7044	0.7012	0.7033	0.6972	0.6857	**0.7074**	0.6576	0.7013	0.7023	0.7051
mfeatmor0	0.9932	**0.9934**	0.9932	0.9932	0.9898	0.9923	0.9896	0.9901	0.9892	0.9919	0.9921	0.9603	0.9887	0.9889	0.9932	0.9881	0.9932
yeast3	0.7836	**0.8006**	0.7929	0.7968	0.7977	0.799	0.7984	0.779	0.799	0.7862	**0.8006**	0.7797	0.7968	0.785	0.7925	0.7907	0.7801
mfeatmor01	0.9585	0.9629	0.961	0.9572	0.9555	0.9562	0.9551	0.9584	0.955	0.96	0.9616	0.9505	0.9533	0.9588	0.9377	0.901	**0.9637**
glass123vs567	0.9076	**0.9172**	0.907	0.9127	0.9034	0.903	0.9016	0.9035	0.907	0.9044	0.9085	0.8573	0.9025	0.9024	0.8943	0.8909	0.9032
parkinsons	0.8064	0.844	0.8315	0.833	0.8426	0.8433	0.8432	0.836	0.844	0.8353	0.821	0.7942	**0.849**	0.8222	0.8349	0.847	0.8361
habermans_survival	0.4767	0.5015	0.4834	0.506	0.4704	0.4676	0.485	0.428	0.4882	0.4356	0.5086	0.4228	0.468	**0.5287**	0.4624	0.4677	0.4263
glass23567vs1	0.8478	**0.8756**	0.8583	0.8544	0.8516	0.8519	0.8493	0.8547	0.8483	0.861	0.827	0.8561	0.8507	0.8237	0.8471	0.8456	0.8583
breast_cancer	0.9564	**0.9598**	0.9566	0.9574	0.9556	0.9558	0.9568	0.956	0.9558	0.9574	0.9541	0.9502	0.9561	0.9547	0.953	0.9575	0.9561
banknote	0.9936	**0.9944**	0.9937	0.9939	0.9936	0.9935	0.9935	0.9933	0.9936	0.9934	0.9932	0.964	0.9936	0.9906	0.9933	NA	0.9933

Bibliography

Ng, Kai Wang, Tang, Man-Lai, and Tian, Guo-Liang. (2011). *Dirichlet and Related Distributions: Theory, Methods and Applications*. 1st ed. Wiley Series in Probability and Statistics, Vol. 895. Wiley, Newark. ISBN: 9780470688199.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.