Exploring New Frontiers in Imbalanced Learning: Data Complexity-Based Solutions

by

ASIF NEWAZ

A thesis report submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
IN
COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering
East West University
Dhaka-1212, Bangladesh.
September, 2024

© 2024 ASIF NEWAZ
All Rights Reserved.

CERTIFICATE OF APPROVAL

The thesis titled, "**Exploring New Frontiers in Imbalanced Learning: Data Complexity-Based Solutions**" submitted by ASIF NEWAZ, Student ID 2022-2-96-004, has been found as satisfactory and accepted as partial fulfillment of the requirement for the degree MASTER OF SCIENCE in COMPUTER SCIENCE AND ENGINEERING on September , 2024.

Board of Examiners:

Dr. Taskeed Jabid (Supervisor)
Associate Professor,
Department of Computer Science and Engineering,
East West University, Dhaka.

Dr. Maheen Islam (Chairperson)
Associate Professor,
Department of Computer Science and Engineering,
East West University, Dhaka.

Declaration

I, Asif Newaz, hereby declare that the work presented in this thesis report is the outcome of the investigation performed by me under the supervision of Dr. Taskeed Jabid, and no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma, except for publication.

Dr. Taskeed Jabid
Associate Professor,
Computer Science and Engineering,
East West University

Asif Newaz
Student No.: 2022-2-96-004

Dedicated to myself .....

Table of Contents

List of Figures

List of Tables

List of Abbreviations

OS	Oversampling
US	Undersampling
CS	Cost Sensitive
OVO	One-vs-One
OVA	One-vs-All
BRF	Balanced Random Forest
IR	imbalance ratio
ROS	Random Oversampling
SMOTE	Synthetic Minority Oversampling Technique
ADASYN	Adaptive Synthetic Minority Oversampling Technique
BL-SMOTE	Borderline-SMOTE
CURE-SMOTE	Combining Clustering Using Representatives (CURE) with SMOTE
MW-SMOTE	majority-weighted SMOTE
IPF	Iterative-Partitioning Filter
ROSE	Random Over-Sampling Examples
G-SMOTE	Geometric SMOTE
RUS	Random undersampling
IRUS	Inverse random undersampling
CBEUS	Clustering Based Evolutionary Undersampling
ENN	Edited Nearest Neighbors
CNN	Condensed Nearest Neighbor
ACO	Ant colony optimization
IHT	Instance Hardness Threshold
SVM	Support Vector Machine
ML	Machine Learning
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
G-mean	Geometric Mean
ROC-AUC	Receiver Operating Characteristic - Area Under the Curve
TPR	True Positive Rate
FPR	False Positive Rate
MCC	Matthews Correlation Coefficient
SLSMOTE	Safe-Level-SMOTE
MWMOTE	Majority Weighted Minority Oversampling Technique
NCL	Neighborhood Cleaning Rule
DBMUTE	Density-based Majority Undersampling Technique
OBU	Overlap-based undersampling
RD	Redundancy Driven
EBUS	Evolutionary Undersampling
EUSBoost	Evolutionary Undersampling Boosting
RF	Random Forest
iBRF	Improved Balanced Random Forest
LVQ-SMOTE	Learning Vector Quantization SMOTE
XGBoost	eXtreme Gradient Boosting
OVA	One-Vs-All
OVO	One-Vs-One
DL	Deep Learning
DRL	Deep Reinforcement Learning
GAN	Generative Adversarial Networks

Acknowledgment

All praise be to Allah, the most beneficent, the most merciful. I would like to express my deepest gratitude to the almighty Allah for His blessings on me. My special thanks go to my supervisor, Dr. Taskeed Jabid, who gave me this opportunity, and without whom this work would not have been possible. Thanks also go to my students Md. Salman Mohosheu, Mr. Asif Ur Rahman Adib, Md. Abdullah al Noman and Mr. Farhan Shahriyar Haq for their contributions. I would also like to acknowledge the continuous support of my family without whom nothing would be possible. A lot of thanks to my friends who stayed beside me in my difficult times.

Asif Newaz
July 2024

Abstract

Class imbalance is a frequently occurring scenario in classification tasks. Learning from imbalanced data poses quite a challenge which has instigated a lot of research in this area. Various techniques have been developed over the years to tackle this problem. These approaches are broadly classified into two categories: Data-level modification and Algorithm-level modification. In data-level modification, the original class distribution in the data is altered through resampling techniques. In algorithm-level modification, the traditional classification algorithms are adjusted to the imbalanced scenarios by changing the cost function and making them cost-sensitive (CS).

A lot of different data resampling and CS techniques have been proposed by researchers in the past decade. To understand their strengths and weaknesses, a comprehensive experimental analysis is first conducted to obtain insights about these techniques. Several limitations have been identified that limit the performance of these approaches. Most of these techniques do not take into consideration data intrinsic characteristics that complicate the learning process. Several data difficulty factors have been identified in some previous studies which are rarely addressed in most cases. Moreover, the application of many of these techniques overfits the data and causes a loss of generalization, producing poor performance while testing. They are also unable to provide well-generalized performance on a wide range of imbalanced scenarios.

In this study, novel strategies have been developed to address these issues. Solutions have been proposed to limit the effects of different data difficulty factors and enhance prediction performance. Moreover, attempts have been made to overcome the shortcomings of the established approaches and obtain better generalization. Three different methods have been proposed in this study. First, a novel data resampling technique that takes into consideration data intrinsic characteristics to effectively balance the dataset. Second, an instance complexity-based CS technique which is an advanced modification to the original CS approach. Third, a hybrid framework combining resampling and CSL.

Rigorous experiments have been conducted on a wide range of imbalanced datasets to validate the performance of the proposed approaches. The results have been evaluated on eight different performance measures and compared with other state-of-the-art techniques used in imbalanced learning. Superior results have been obtained from the proposed techniques on different imbalanced scenarios. The results demonstrate the efficacy of the proposed models in learning from imbalanced data.

To conclude, this research delineates new trajectories in the field of the imbalanced domain. New approaches have been proposed that introduce fresh perspectives and directions in imbalanced learning. The proposed strategies are remarkably successful, ensuring well-generalized performance when addressing imbalanced data.

Keywords: Class imbalance, Class overlap, Cost-sensitive learning, Data difficulty factors, Empirical study, Multiclass classification, SMOTE.

Chapter 1

Introduction

1.1 Overview

Learning from imbalanced data is a major challenge in classification tasks. Real-world datasets often come with different degrees of imbalance. One class is usually underrepresented (minority class) compared to the other class (majority class). In applications such as medical diagnosis, fault detection, and fraud identification, class imbalance is quite prevalent. Traditional classification algorithms are not suitable to deal with such scenarios. Standard classifiers are designed in such a way that they are trained to minimize the number of misclassifications, irrespective of the class. Therefore, if one class is underrepresented in the data, the performance gets biased towards the majority class. The problem intensifies if the disparity between the classes is larger. There are other data intrinsic characteristics that further complicate the scenario. The classifier might completely overlook the minority class samples and classify all test samples as the majority class. However, identifying the minority class samples correctly is often imperative. In the healthcare context, misdiagnosing a cancer patient as normal can have severe consequences. Therefore, it is essential to take necessary measures to tackle the class imbalance in order to achieve satisfactory performance. This has attracted a lot of attention from researchers over the years, and a variety of techniques have been proposed to tackle the problem [1].

The techniques used in imbalanced learning tasks can be broadly classified into two categories:

- Data level approach
- Algorithmic level approach

The data level approach refers to resampling techniques where the original class distribution in the data is modified [2]. This is done through oversampling (OS) or un-

dersampling (US). In OS, new minority-class samples are generated to increase their presence. This is achieved by duplicating the existing samples or synthesizing new samples through some heuristics. On the other hand, in US, instances from the majority class are removed. This can be done randomly or heuristically. The goal is usually to balance the dataset. Recent investigations suggest it is even more important to reduce the class overlapping in the process [3]. Class overlap refers to the phenomenon where data points from different classes are not distinctly separable in the feature space, causing them to intermingle. This overlap can lead to misclassification and reduce the accuracy of classification algorithms, as the boundaries between classes are blurred [4].

In algorithmic-level approaches, the original classification algorithm is modified to adapt to the imbalanced domain scenario. This is achieved by changing the cost function to handle the class imbalance directly [5]. Higher misclassification costs are assigned to the minority class instances to make the algorithm more sensitive to those errors. During training, the model learns by trying to reduce the overall misclassification cost. Assigning higher weight to the minority-class misclassifications shifts the bias from the majority class. This way, the algorithm is made cost-sensitive (CS). This approach is classifier dependent as different algorithms use different learning procedures.

These two categories of techniques adopt two different approaches to deal with the imbalanced scenario. Both of these have been very successful in addressing the class imbalance problem and are widely used in many applications [6–8].

1.2 Problem Statement

While a plethora of approaches have been developed for imbalanced classification tasks, a very limited amount of research has been conducted to identify the issues that make imbalanced classification so difficult. Only a limited number of studies have analyzed the efficacy of the established techniques on a wide variety of imbalanced datasets. This has created a research gap in this domain and this study aims to fill that void.

Class imbalance in the data is typically held responsible for the decline in performance observed in standard classifiers. Consequently, conventional resampling methods aim to rectify this issue by equalizing the class distribution to alleviate the problem. However, recent studies, including our own investigations, have revealed that class imbalance is not the primary factor contributing to this issue [9]. There are other data intrinsic characteristics that exert a greater influence on the challenges encountered in learning from imbalanced data. These include –

- Class overlapping
- Presence of noisy samples
- The rarity of the samples
- Small disjuncts

These factors need to be addressed to improve the prediction performance. However, the traditional resampling techniques as well as CS approaches do not take these issues into consideration. Consequently, they suffer from a severe drop in performance in different imbalanced scenarios. For instance, many of the popular approaches do not perform well in highly imbalanced datasets due to the limitations of their design. These established approaches also introduce other issues while modifying the algorithm or the dataset (observed during experimentation in this study). These are summarized as follows –

- OS techniques introduce noisy samples in the data while generating synthetic minority class samples. These noisy instances do not represent the actual minority class and cause overfitting.
- OS techniques also increase overlapping with the opposite class, resulting in a loss of generalizability.
- US techniques can cause loss of valuable information if too many samples are removed.
- US techniques do not increase the presence of minority class samples in the dataspace which is crucial for accurate identification of the positive cases.
- Many of these techniques only work well on low imbalances but fail in highly imbalanced datasets.
- CS approaches penalize all the minority class samples equally. However, not all the samples offer the same level of learning difficulty. Indiscriminately penalizing all the samples can create some unusual deformation of the decision boundary and consequently, cause a higher number of misclassifications of the majority class instances.

The traditional approaches used in the imbalanced domain do not adequately take into account all the different data difficulty factors that are responsible for the intricacies encountered in imbalanced learning. Additionally, these methods tend to introduce additional complexities that lead to overfitting and a decrease in generalization. It is crucial to consider all these aspects when devising new approaches, and this research aims to explore these considerations thoroughly.

1.3 Objectives

The aim of this research is to develop new and effective strategies for imbalanced learning. In that regard, an extensive experimental study and literature review are first conducted to identify the major issues and find the shortcomings of the established approaches. The algorithms developed in this study address all the data difficulty factors while also alleviating the issues associated with popular methods. In summary, this research has the following objectives –

- To thoroughly investigate the performance of established methods used in imbalanced learning on a wide range of imbalanced scenarios to identify their strengths and weaknesses.
- To thoroughly review the literature on imbalanced learning, especially those that raise concerns regarding typical approaches and their limitations.
- To devise novel strategies capable of mitigating the challenges posed by data complexity, thus enabling the attainment of satisfactory performance.
- To develop new methodologies capable of surpassing the constraints of existing approaches, thereby delivering enhanced performance.
- To develop novel approaches that can provide well-generalized performance on a wide range of imbalanced datasets.

1.4 Contributions

The main contributions of this research are as follows.

- A detailed experimental study has been carried out on a wide range of imbalanced datasets to observe the performance of popular and state-of-the-art techniques used in imbalanced learning. A critical discussion on these approaches has been provided. Through an in-depth analysis, several major limitations of the established approaches have been identified. The efficacy of these approaches on a range of imbalanced scenarios has been discovered. The primary factors contributing to the challenges of learning from imbalanced data have been determined. How different sampling techniques deal with these factors have been singled out. Overall, through this empirical study, a comprehensive analysis of the established approaches in the imbalanced domain has been provided. *This work has been published at the 2024 International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), Dhaka, Bangladesh [10].*

- A novel data resampling methodology has been proposed. This approach is aimed at minimizing the effect of the data difficulty factors while also addressing the limitations of the popular approaches. *This work is currently under review in the Knowledge and Information Systems journal.*

- Based on the developed resampling methodology, a novel ensemble algorithm has been proposed. This ensemble approach is a modified and improved version of the original Balanced Random Forest (BRF) classifier. *This work has been published at the 2024 35th Conference of Open Innovations Association (FRUCT), Finland [11].*

- A novel instance complexity-based CS framework has been proposed. While traditional CS approaches penalize all the minority-class instances equally, in the proposed approach, instances are weighted according to their difficulty level. This provides a more plausible weighting mechanism with enhanced performance. *A paper on this work is currently under review at the IEEE International Conference on Data Mining (ICDM).*

- Hybridization between data resampling and Cost-sensitive learning (CSL) can be quite effective in handling class imbalances. This approach has been applied to a real-world problem of predicting complications of myocardial infarction within several hours of hospitalization. Improved prediction performance has been observed compared to other popular approaches. *The work has been published in Informatics in Medicine Unlocked journal [12].*

- A novel multilevel decomposition strategy has been proposed for multiclass classification. The proposed approach provides improved performance over traditional One-vs-One (OVO) or One-vs-All (OVA) decomposition techniques. Here, instead of unsystematically binarizing the dataset to handle multiclass scenarios, a sophisticated decomposition methodology has been adopted. *A paper on this work is currently being prepared.*

1.5 Thesis Outline

The remainder of the article is organized as follows.

- Chapter 2 provides a detailed discussion of the imbalanced learning problem. The methodologies used in learning from imbalanced data have been reviewed. A thorough discussion on evaluation metrics for imbalanced data has been presented. Findings from some recent investigations on imbalanced learning have also been discussed.

- In Chapter 3, a comprehensive analysis of the well-established methods in the imbalanced domain has been presented. The strengths and weaknesses of the methods have been discussed. The complexities the classification algorithms face when dealing with skewed class distribution have also been described here.
- Chapter 4 introduces a novel data resampling methodology that simultaneously focuses on reducing class overlapping and class imbalance for improved performance.
- Chapter 5 presents a novel instance-level CS framework that weights instances according to their complexity.
- Chapter 6 details a new hybrid framework between data resampling using the SMOTE algorithm and CSL.
- Chapter 7 concludes this article with a summary of the thesis, its limitations, and future research prospects.

Chapter 2

Research Background

This chapter presents a detailed introduction to the imbalanced classification problem. The methodologies used in the imbalanced domain are categorically presented and discussed. A critical analysis of the appropriate metrics for performance evaluation has been provided. Some of the recent works in this domain from other researchers have also been reviewed.

2.1 Foundations of Imbalanced Learning

Imbalanced data refers to a dataset in which the classes are not represented equally [2]. In other words, one class (or a few classes) has significantly more instances than the other(s). This imbalance can lead to biased models that perform well on the majority class but poorly on the minority class(es). Imbalanced data is common in many real-world applications, such as fraud detection, medical diagnosis, and rare event prediction.

Class imbalance poses quite a challenge in predictive modeling. As the performance gets biased towards one class (majority class), the classifier fails to correctly identify instances from the other class(es), which are usually rare cases. However, it is often more desirable to identify these rare instances correctly as they typically represent positive cases. Misidentifying these important examples or providing biased predictions is not acceptable for a reliable prediction framework. For instance, in a fraud detection task, the data usually contains hundreds of thousands of normal transactions compared to only a few hundred fraudulent transactions. Now, the goal of the prediction framework is to capture these fraudulent transactions and warn the system. However, a traditional classifier trained on such skewed data usually becomes biased and predicts almost all the transactions as normal, failing to identify the fraudulent cases. The classifier fails to serve its actual purpose. This is unacceptable and appropriate measures need to be adopted to reduce the bias.

The class imbalance scenario has been illustrated in Fig. 2.1. As can be seen from

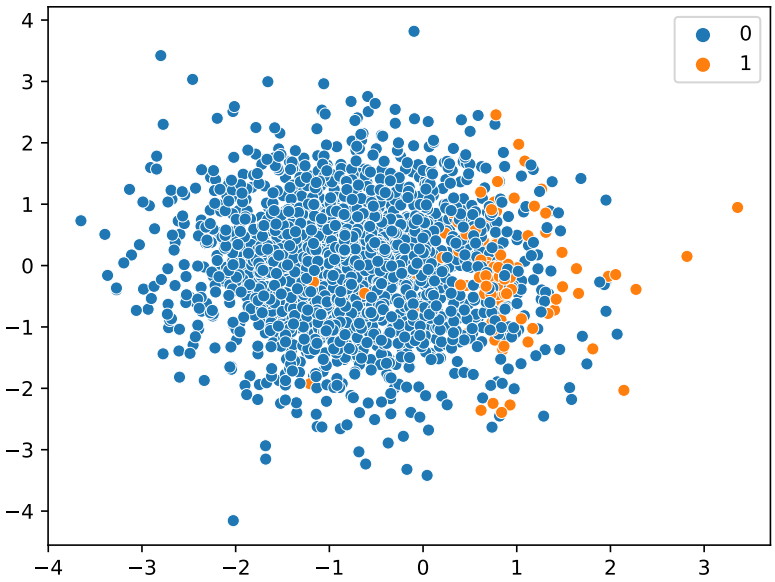

Figure 2.1: Class imbalance in data.

the figure, the majority of the instances belong to class 0 while only a few instances are from class 1. This skewness in the data biases the learners as the minority class becomes overshadowed by the presence of a large number of instances from the opposite class. Many of the minority-class instances are seen as noisy samples by the classifier and ignored. Consequently, the classifier is unable to accurately differentiate between instances of the two classes, resulting in misclassification and a decline in overall performance.

2.2 Strategies for Imbalanced Learning

Many different strategies have been developed to address the issue of class imbalance [13]. Since traditional classifiers do not perform well on imbalanced data, the idea is to either modify the dataset or adjust the algorithm to handle this issue effectively. These techniques are broadly classified into two categories. Data-level and algorithmic-level. They are discussed in detail below.

2.2.1 Data Level Modification

Data-level modification refers to changing the original number of instances in the classes. This is done by generating new minority-class instances or eliminating instances from the majority class. This is commonly referred to as 'sampling' and has become a standard data preprocessing technique in the case of imbalanced data. Re-

searchers have proposed many different sampling techniques, which can be broadly classified into four groups. They are -

- Oversampling (OS)
- Undersampling (US)
- Hybrid Sampling
- Ensemble

Oversampling refers to generating new minority class samples using the existing ones. Undersampling refers to eliminating samples from the majority class. These can be done heuristically or non-heuristically. While non-heuristic approaches are simple and fast, they can cause overfitting or loss of information. Heuristic approaches, on the other hand, are aimed at ensuring the quality of the resampled dataset by strategically generating new minority class samples or carefully removing majority class samples from the original data.

Oversampling (OS)

The idea behind OS is to generate new minority-class instances to increase their presence as well as to reduce the imbalance ratio (IR). This is illustrated in Fig. 2.2. Various OS techniques have been proposed over the years. The simplest one is Random Oversampling (ROS) where samples are just duplicated to balance the dataset. This is a non-heuristic approach and does not add any new information. A more advanced heuristic approach is the Synthetic Minority Oversampling Technique (SMOTE) developed by Chawla et al. [14]. Here, new synthetic samples are generated using interpolation. This is the most renowned sampling technique used in the imbalanced domain.

The way SMOTE works is as follows. Given a minority class example, SMOTE selects its nearest neighbors (typically using Euclidean distance). It then generates synthetic examples by interpolating between the selected example and its neighbors.

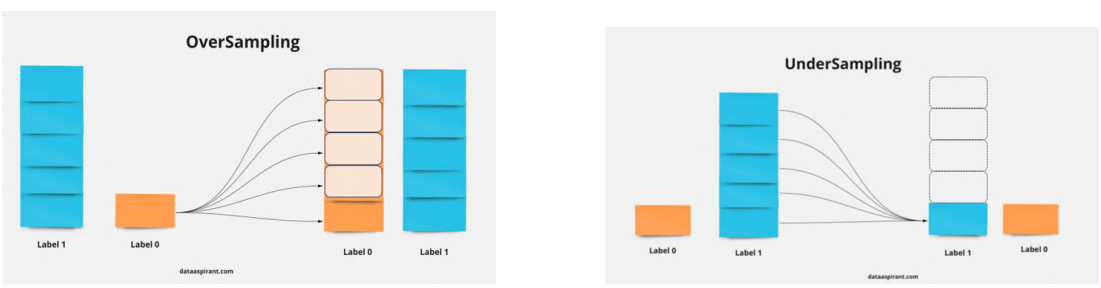

Figure 2.2: Oversampling and Undersampling

The synthetic examples lie along the line segments connecting the original example and its neighbors in the feature space. By doing so, SMOTE creates new instances that represent variations within the minority class. Due to its simplicity and robustness, SMOTE has become a standard benchmark for learning from imbalanced data. The sample generation process of SMOTE is illustrated in Fig. 2.3.

The popularity of the approach has led to the development of numerous variations of the technique [15]. SMOTE, however popular, has its limitations. These variants attempt to address those issues and improve performance. For instance, ADASYN which stands for Adaptive SMOTE, takes an adaptive approach [16]. It focuses on minority instances that are difficult to classify correctly, rather than oversampling all minority instances uniformly. ADASYN generates more synthetic samples for minority instances closer to the decision boundary, creating synthetic samples in challenging areas of the feature space. There are around a hundred variations of the original SMOTE algorithm. An empirical study on 85 such variants was conducted by Kovacs et al. [17]. The authors identified some of the best-performing OS techniques. They also pointed out that there are no major variations in performance among these variants.

A summary of some of the most popular and recent OS techniques is provided below.

- **Borderline-SMOTE (BL-SMOTE)**: Extension of SMOTE focusing on borderline samples as they are more likely to be misclassified [18]. Synthetic samples are generated only near the decision boundary.

- **Safe-Level-SMOTE**: Extension of SMOTE that assigns a safety level to the minority class samples based on its nearest neighbors [19]. New samples are created only in the safe positions. It improves upon SMOTE by considering a "safe level" for each minority instance. By synthesizing more instances around larger safe levels, this approach achieves better accuracy than SMOTE.

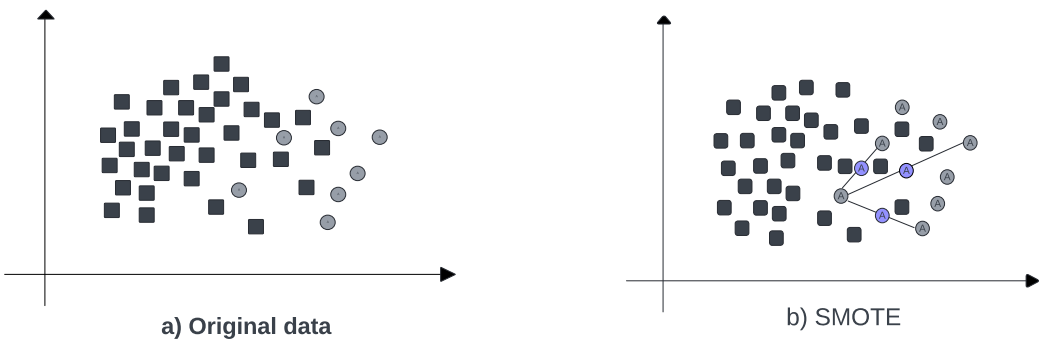

Figure 2.3: Synthetic sample generation using SMOTE

- **DBSMOTE**: Extension of SMOTE which utilizes the DBSCAN clustering algorithm to improve minority class samples detection rate. It generates synthetic instances along the shortest path from each positive instance to a pseudo-centroid of a minority-class cluster [20]. As a result, these newly generated instances are dense near the centroid and are sparse far from the centroid.

- **CURE-SMOTE**: Combining Clustering Using Representatives (CURE) with SMOTE algorithm [21].

- **MWSMOTE**: Focusing on the shortcomings of the SMOTE algorithm, this majority-weighted minority oversampling technique attempts to improve the sample selection and generation scheme [22]. It identifies difficult minority class instances that are hard to classify correctly. It assigns weights to these hard-to-learn instances based on their distance from the majority class instances.

- **SMOTE-IPF**: It adds a new element to SMOTE, an iterative ensemble-based noise filter called Iterative-Partitioning Filter (IPF) [23]. This helps overcome the problems produced by noisy and borderline examples in imbalanced datasets.

- **ROSE**: ROSE (Random Over-Sampling Examples) is a bootstrap-based technique [24]. It handles both continuous and categorical data by generating synthetic examples from a conditional density estimate of the two classes. Unlike simple ROS, ROSE provides a smoothed bootstrap approach, creating a synthetic sample of data. This is done by adding small amounts of noise, ensuring the synthetic samples are similar to real ones.

- **G-SMOTE**: It generates synthetic samples in a geometric region of the input space, around each selected minority instance [25]. While in the basic configuration, this region is a hyper-sphere. G-SMOTE allows its deformation to a hyper-spheroid.

- **NEATER**: It proposes a filtering method of the oversampled data using a non-cooperative game theory [26]. Oversampling creates noisy samples in the process that are eliminated in this approach.

Undersampling (US)

In US, samples from the majority class are removed to reduce the IR. This has been illustrated in Fig 2.2. The simplest approach is to eliminate the samples randomly without any consideration. This approach is known as Random Undersampling (RUS). While this method is simple and fast, it can cause a loss of valuable information. To avoid such complications, more strategic US approaches have been developed over the

years. These approaches use different heuristics to select samples for removal. Some of these approaches are based on the simple nearest-neighbor rule, while others utilize evolutionary algorithms to find instances for removal. Again, some of these approaches aim to balance the class distribution, while others try to reduce class overlapping by eliminating instances from the majority class.

A summary of some of the most popular and recent US techniques is provided below. The sample elimination by RUS is illustrated in Fig. 2.4.

- **IRUS**: This inverse random undersampling technique utilizes the bagging method where the imbalance between the classes is reversed in different subsets of the data [27].
- **CBEUS**: This algorithm integrates the clustering method (k-means clustering) with the genetic algorithm to remove majority-class samples that are far away from the centroid of each group [28].
- **Tomek-link**: This is a data cleaning approach. This algorithm first identifies Tomek links in the data. A Tomek Link exists between two samples from different classes if they are each other's nearest neighbors. In this method, the instance in the tomek-link belonging to the majority class is removed.
- **ENN**: Edited Nearest Neighbors, is a US technique used to remove noisy or borderline instances from the majority class [29]. It is particularly useful when the dataset contains overlapping clusters or when the majority class has instances that are too close to the minority class. It uses the nearest neighbor rule to select instances.
- **CNN**: Condensed Nearest Neighbor (CNN) is a technique used in imbalanced learning for data reduction, particularly for large datasets with many instances [30]. It significantly reduces the size of the dataset by selecting a subset of

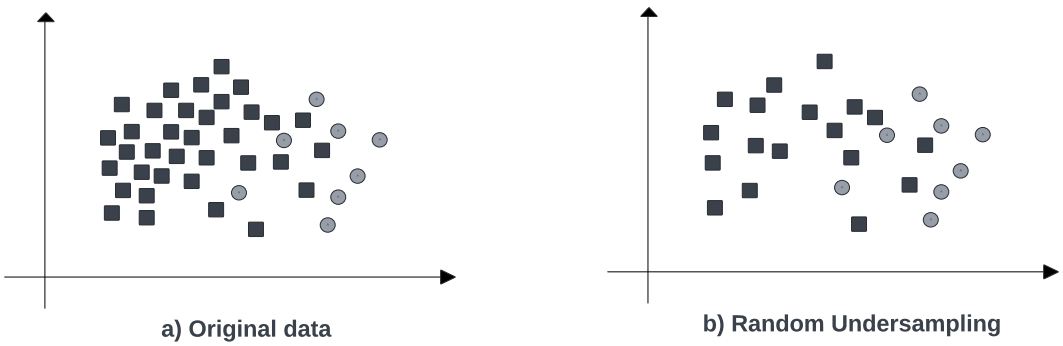

Figure 2.4: Majority class sample elimination using RUS

representative samples while maintaining the discriminatory information. The idea behind this approach is to eliminate the examples from the majority class that are distant from the decision boundary and therefore, can be considered less relevant for learning.

- **ACOSampling**: This is a meta-heuristic US approach that uses ant colony optimization (ACO) to find a sub-optimal subset of the majority class for classification [31].

- **IHT**: Instance hardness is a measure of how challenging a particular instance is for a given learning algorithm to predict correctly [32]. It reflects the uncertainty or ambiguity associated with individual instances. Instance hardness thresholding involves setting a threshold on the hardness measure to filter out instances below a certain hardness level.

Hybrid Sampling

Hybridization between OS and US is also a plausible solution, especially when the IR is high. The goal is to have a balance between the two techniques. It limits the generation of too many synthetic instances to avoid overfitting. Similarly. it also reduces the elimination of too many majority-class instances, lowering the loss of information. This way, such a balanced hybridization can outperform the other techniques. Several such hybridizations have been proposed and a summary of these approaches is presented below.

- **SMOTE-Tomek**: In this hybrid method, the Tomek-link approach is first used for undersampling [33]. SMOTE is then used for oversampling. However, the number of samples removed using this method is very limited. So, the performance does not vary much from SMOTE.

- **SMOTE-ENN**: In this method, ENN is first used for undersampling, followed by SMOTE [33]. ENN provides better cleaning than Tomek-link. More samples are removed by ENN. Then SMOTE is used to generate the necessary number of samples to balance the dataset.

- **Random Balance**: Here, US and OS techniques are combined with random sampling ratios on different subsets [34]. It uses ensemble methods for obtaining different subsets. The approach was later extended for multiclass imbalanced scenarios [35].

- **SMOTE-CNN**: Here, SMOTE is merged with the CNN approach [36]. CNN provides a large reduction in the number of majority-class instances. Thereby, this approach attains a better balance between OS and US.

Ensemble Methods

Ensemble algorithms are more robust compared to standard classifiers. Multiple weak learners are combined in ensembles to obtain better performance and reduce overfitting and bias. Bagging and boosting are two popular ensemble methods. Bagging uses the bootstrapping technique to produce multiple subsets of the original data, whereas boosting uses an instance weighting mechanism. While they are capable of reducing overfitting and bias, they remain susceptible to the imbalanced classification problem (the data the algorithms are trained on remains skewed).

Sampling approaches can be integrated into the ensemble learning frameworks. In bagging, each bootstrap subset can be balanced using a sampling method. This mitigates the class imbalance problem. However, it does not solve the class overlap problem. Similarly, in boosting, sampling techniques are applied to the data in each boosting iteration. Different such ensemble algorithms have been proposed and a summary is provided below.

- **OverBagging**: This method uses ROS to balance each bootstrap subset generated in the bagging process. The remaining process is the same as before.
- **SMOTE-Bagging**: Similar to the previous method, this approach uses SMOTE to balance the bootstrap subsets.
- **BRF**: This approach uses RUS to balance the subsets [37]. The RF framework is then used for prediction. Balancing the subsets and using an ensemble also reduces the information loss as many such subsets are produced.
- **RUSBoost**: This approach uses RUS to balance the data. It uses the boosting (AdaBoost) framework for learning.

Many other variations of the ensemble approaches for imbalanced learning have been proposed. For instance, SMOTE-Boost, Under-Bagging, Easy Ensemble [38], Hard Ensemble [39], EUSBoost [40], IMCStacking [41], etc.

2.2.2 Algorithmic Level Modification

In the algorithmic-level approach, the original classification algorithm is modified to adapt to the imbalanced domain scenario. This is achieved by modifying the cost function to account for class imbalance directly. Specifically, higher misclassification costs

are assigned to instances of the minority class, making the algorithm more sensitive to errors involving those instances. During training, the model focuses on reducing the overall misclassification cost. By assigning greater weight to misclassifications of the minority class, the bias is shifted away from the majority class. This results in a cost-sensitive (CS) algorithm. This approach is dependent on the classifier, as different algorithms utilize different learning procedures.

Cost-Sensitive Learning

In Cost-Sensitive Learning (CSL), a specific penalty is assigned to misclassifications of minority-class instances. Standard classifiers typically use a 0-1 loss function to calculate the cost, where a correct classification scores 0 and an incorrect one scores 1. This error-driven (ED) approach assumes an even class distribution in the dataset. However, when data is imbalanced, this method performs poorly, especially in terms of sensitivity (the accuracy of minority class predictions). In many applications, correctly classifying minority-class instances, often representing positive cases, is crucial. To address this, the concept of a cost-driven classifier is introduced, which employs asymmetric misclassification costs. By assigning a higher cost to misclassifications of minority-class instances compared to those of the majority class, the algorithm is compelled to prioritize learning these instances correctly, thus reversing the bias. This approach is particularly effective for imbalanced datasets.

The implementation of cost-sensitive algorithms relies on a cost matrix, as shown in Table. 2.1. In this matrix, C_1 represents the penalty for errors in minority class predictions, while C_2 represents the penalty for errors in majority class predictions. Increasing the value of C_1 enhances the recall or sensitivity score. Typically, C_2 is related to the specificity score and is usually set to 1. Assigning a higher weight to majority-class instances can negatively impact the performance of the minority class, so it is generally avoided. The penalty values can be chosen arbitrarily or optimized using search algorithms.

The effect of modifying the weights of the instances on the decision boundary for the Support Vector Machine (SVM) classifier is illustrated in Fig 2.5. As shown in the figure, assigning a higher weight to certain instances compels the classifier to place greater emphasis on correctly classifying those points, thereby altering the original

Table 2.1: Cost Matrix

	Predicted True	**Predicted False**	
Actual True	0	C_1	**Minority Class**
Actual False	C_2	0	**Majority Class**

decision boundary.

2.3 Performance Evaluation

Evaluating the performance of machine learning (ML) models is crucial to understanding their effectiveness, reliability, and suitability for a given task. Different metrics and methods are used depending on the type of problem (classification, regression, clustering, etc.). Accuracy is the most commonly used metric for classification tasks. However, when the data is imbalanced, special consideration is required as traditional measures of performance become biased. Let's look at the example below.

Say, in a dataset, there are 10000 samples. 9900 of them belong to the negative class (majority) and the remaining 100 belong to the positive class (minority). Now, if the ML model predicts all the instances as negative, it will still be correct for 9900 cases, even though all the positive cases were predicted wrongly. The accuracy of the prediction framework would be 99%, which is extremely high while the model is actually failing to distinguish positive cases from negative ones. This type of scenario occurs frequently when the data is imbalanced and therefore, an imbalanced domain requires some special attention when it comes to performance evaluation.

The performance measures of classification models are defined based on a confusion matrix. This is illustrated in Fig 2.6. There are four elements in a confusion matrix. They are as follows.

- True Positive (TP): Predicted positive, actual positive.
- True Negative (TN): Predicted negative, actual negative.
- False Positive (FP): Predicted positive, actual negative.
- False Negative (FN): Predicted negative, actual positive.

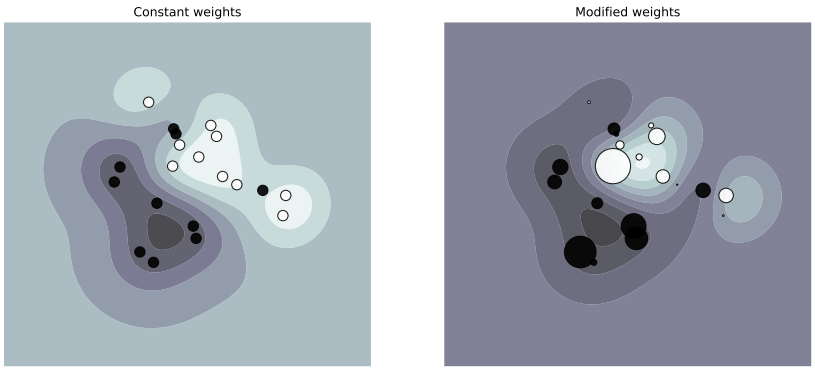

Figure 2.5: Effect of modifying weights of the instances on the decision boundary.

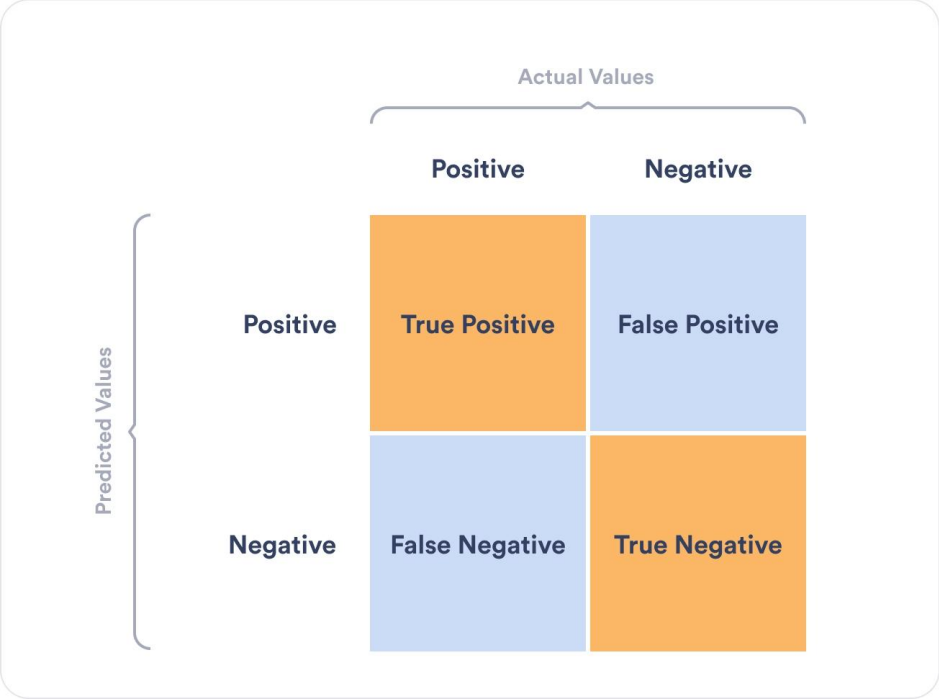

Figure 2.6: Confusion Matrix.

Different performance measures are defined using these four elements. Some of these measures are class-dependent, while others are composite metrics and more robust. Not all of them are suitable for performance measurements in imbalanced data. These metrics are briefly described below.

2.3.1 Evaluation Metrics

- **Accuracy:** The ratio of correctly predicted instances to the total instances. This metric is not suitable in imbalanced cases as the measures get biased by the majority class.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

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

- **Recall (Sensitivity or True Positive Rate):** The ratio of correctly predicted positive observations to all observations in the positive class. This is a class-specific metric that shows the accuracy of the model in predicting positive cases. While this measure is useful in understanding how accurately the model can identify the positive cases, it does not show the entire spectra of performance on the whole dataset.

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

A sensitivity score of 0.8 in a dataset with only 100 minority-class samples implies that the model misclassified 20 positive cases.

- **Specificity (True Negative Rate):** The ratio of correctly predicted negative observations to all observations in the negative class. This is another class-specific metric that shows the accuracy of the model in predicting negative cases. While this measure is useful in understanding how accurately the model can identify the negative cases, it does not show the entire spectra of performance on the whole dataset.

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

A specificity score of 0.8 in a dataset with 10000 majority-class samples implies that the model misclassified 2000 positive cases.

- **Precision (Positive Predictive Value):** The ratio of correctly predicted positive observations to the total predicted positives. Although precision is very popular in performance measurements, in the case of imbalanced data, this metric also displays biased performance.

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

- **F1 Score:** The weighted average of Precision and Recall. This is a composite metric that considers both precision and recall values. It only becomes high when both precision and recall scores are good.

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

- **Balanced Accuracy:** The arithmetic average of specificity and recall. This is a composite metric that considers both specificity and recall values. This shows the average performance of the model in both classes. This is much better than the accuracy score as it considers the performance of individual classes.

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

- **Geometric Mean (G-mean):** The geometric mean of specificity and recall. This is another composite metric that considers both specificity and recall values. This

portrays a more detailed picture of the performance of the model compared to sensitivity and specificity alone. This is also much better than the simple balanced accuracy score. Considering geometric mean allows higher penalization in a drop in performance. If either sensitivity or specificity reduces, the g-mean score reduces.

$$\text{G-Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$$

- **ROC-AUC:** The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) is a performance measurement for classification problems at various threshold settings. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold levels. The AUC (Area Under the Curve) represents the degree or measure of separability, indicating how well the model distinguishes between classes.

$$\text{AUC} = \int_0^1 \text{TPR}(t)\, d(\text{FPR}(t))$$

  This integral represents the area under the ROC curve from FPR $= 0$ to FPR $= 1$. The value of AUC ranges from 0 to 1, with 1 indicating perfect classification and 0.5 indicating a model with no discriminative power (equivalent to random guessing).

- **MCC:** The Matthews Correlation Coefficient (MCC) is a metric used to evaluate the quality of binary classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure, even if the classes are of very different sizes. This metric directly takes into account the actual number of misclassifications, unlike other composite metrics.

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

2.3.2 Discussion on Appropriate Metrics for Imbalanced Data

Machine learning algorithms often excel at predicting instances from the majority class but tend to perform poorly on the minority class. Consequently, traditional performance metrics like accuracy can be misleading because they do not account for class distribution. Sensitivity and specificity are two class-specific metrics that measure the performance accuracy for the minority and majority classes, respectively. However, they only reflect the performance of a particular class, making it challenging to capture the overall performance spectrum.

The geometric mean of sensitivity and specificity combines these metrics into a single value representing the algorithm's overall performance. This metric is particularly useful for evaluating performance on imbalanced datasets, as bias towards any particular class results in a poor G-mean score. While G-mean reflects both sensitivity and specificity, it does not illustrate the trade-off between them. Furthermore, it does not account for the actual number of misclassifications made by the model. For instance, a sensitivity score of 0.8 is not equivalent to a specificity score of 0.8 in skewed datasets. A sensitivity score of 0.8 with only 100 minority-class samples implies that the model misclassified 20 instances. Conversely, a specificity score of 0.8 with 10,000 majority-class samples indicates 2,000 misclassifications. This significant difference in misclassifications must be considered when dealing with imbalanced data. If the imbalance ratio is high, a specificity score of 0.8 would indicate a substantial number of misclassifications in the majority class. A classifier might achieve a high sensitivity score by correctly classifying a small number of minority class samples and a similar specificity score by misclassifying a large number of majority class samples. Nevertheless, the G-mean score might still be high, failing to capture the inherent issue [42].

The balanced accuracy metric is more biased compared to the g-mean score. A sensitivity score of 0 and a specificity score of 100 would still provide a balanced accuracy score of 50. Whereas, the g-mean score would be 0. F1-score is the harmonic mean of sensitivity and precision. However, this metric is also biased for similar reasons.

In the ROC-AUC score, a scalar value representing the area under the ROC curve allows for efficient performance comparison between different approaches. While this metric is unaffected by data skewness, it may obscure poor performance [43]. Additionally, the performance difference between the two approaches in terms of ROC-AUC may be minimal, making comparison challenging.

The MCC score is a more robust measure of performance in classification tasks [44]. It considers all four confusion matrix parameters and only provides a high score when the classifier performs well across all categories. The MCC score will drop if too many misclassifications are made. However, as the dataset becomes more imbalanced, the behavior of MCC becomes skewed and nonlinear with respect to the TP and TN values [45]. Nevertheless, among the composite metrics, only MCC considers the actual number of misclassifications, thereby providing an efficient performance measure.

The performance of the techniques on imbalanced data cannot be adequately represented by a single metric. Therefore, as suggested in previous literature [46], we do not rely on a single metric to evaluate the performance. Rather, four composite metrics were considered for evaluation.

2.4 Related Study

In this section, we review some of the recent studies conducted on imbalanced learning. We focus on those articles that provided a critical review of the approaches used in the imbalanced domain.

While a lot of different techniques have been developed to tackle the imbalanced classification problems, there are still lots of issues such as overfitting, poor performance on highly imbalanced data, and large variations in the prediction [47]. This necessitates an investigation into the underlying issues of imbalanced learning. It is important to understand what makes imbalanced classification so difficult, what factors limit the performance, what the main obstacles are, and how established approaches address these challenges. We try to draw insights from already published articles. Unfortunately, there are very few articles investigating the original issues. We summarize the findings below.

Dudjak et al. [9] identified several intrinsic data characteristics that make imbalanced classification challenging. These include the imbalance ratio, class overlap, rarity of samples, presence of noisy samples, and small disjuncts. A higher degree of imbalance naturally complicates the learning process. However, IR is not the only source of learning difficulty. Other factors also have a significant effect on the performance of different techniques.

Vuttipittayamongkol et al. [48] provided a detailed discussion on the effect of class overlapping. They identified class overlapping as the primary factor affecting classifier performance, suggesting that the impact of class imbalance is significantly influenced by the presence of class overlap. They demonstrated that a higher degree of overlapping can severely degrade performance, even when the data has only a small imbalance. They claimed that IR is not the main contributing factor, but rather a supporting factor. If there is no overlap, classifiers can perform extraordinarily well even in large imbalanced scenarios. Whereas, with increased overlapping, the task becomes more complicated with an increase in IR.

To demonstrate how different algorithms address the class overlapping issue, the authors in [48] categorized the sampling techniques into two types: Class distribution-based and class overlap-based. The class distribution-based approaches focus on balancing the data. On the contrary, class overlap-based approaches try to reduce overlapping or focus on the overlapping regions. In US-type methods, samples are strategically removed from the overlapping regions to reduce the class overlap with the minority-class instances. This category of techniques does not usually focus on balancing the data. The authors did not conduct any empirical study or analysis on the performance of these algorithms related to overlapping or how these approaches tackle

the overlapping issue (conducted in this thesis). An overview of the existing methodologies based on this proposed categorization is presented in Table 2.2.

Table 2.2: Categorization of sampling techniques

Category	Type	Methods
Class overlap-based	Oversampling	ADASYN [16], SMOTE-IPF [23], Borderline-SMOTE [18], SLSMOTE [19], MWMOTE [22]
	Undersampling	ENN [29], NCL [49], DBMUTE [50], OBU [51], RD Tomek-link [52]
	Ensemble	Hard Ensemble [39], EVINCI [53]
Class distribution-based	Oversampling	SMOTE [14] and most of its variants (DB-SMOTE [20], LEE [54], Polynom-fit-SMOTE [55], etc.), ROS
	Undersampling	RUS, IRUS [27], EBUS [56], Clustering-based undersampling [57]
	Ensemble	SMOTE-Bagging [58], Over-Bagging, SMOTE-Boost [59], RUSBoost [60], BRF [37], EUSBoost [40]

Mercier et al. [4] analyzed the degradation of performance of classifiers in different imbalanced contexts and proposed a way of measuring class overlapping called 'degOver'. There are several other proposed ways of quantifying class overlapping and a detailed taxonomy of class overlap complexity measures is presented in this article [3].

In a recent article, Santos et al. [61] advocated for a unified view of class overlapping and class imbalance and suggested that the presence of one element can enhance the impact of the other. The authors suggested developing new techniques capable of addressing both of these issues simultaneously, as existing approaches typically address only one of these challenges at a time.

In a different study, Tarawneh et al. [62] pointed out that oversampling the data using SMOTE and similar approaches generates noisy samples that do not accurately represent the minority class. This leads to overfitting and over-optimistic results. The authors further suggested to stop using oversampling techniques due to this reason. However, they did not provide any direction as to how to tackle the imbalanced scenario. OS is crucial to increase the presence of minority-class samples in the feature space without which performance improvement becomes limited.

To get rid of the noisy samples generated in the oversampling process, some modifications to the SMOTE algorithm have been proposed. These approaches usually apply a filter method to eliminate the noisy samples from the data before training. This includes methods like SMOTE-IPF [23], DBMIST-US [63], SMOTE-ENN [33], etc.

In the case of tabular data, the number of minority class samples available is usually quite limited. Especially in medical-related fields, samples are very rare. This makes it difficult to employ deep learning techniques such as Generative Adversarial Networks (GAN) or Deep Reinforcement learning (DRL) as they require a higher number of samples for training [64].

Most of these studies are focused on sampling techniques. A detailed review of different CS methods is presented in this article by Petrides et al. [65]. The concept of incorporating class overlapping and other data complexity factors into CS frameworks is a relatively new idea and has not been attempted to the best of our knowledge. In this thesis, a detailed experimental study is first conducted on the performance of different approaches on a wide range of imbalanced data. The findings are consistent with the concepts discussed in the aforementioned literature. Additionally, several new issues were identified and are presented in the following chapter.

Chapter 3

Efficacy Analysis of Different Techniques used in Imbalanced Learning

In this chapter, we first discuss the complexities of imbalanced classification tasks. Different data intrinsic characteristics are responsible for the intricacy. An experimental study was conducted on 84 imbalanced datasets to evaluate the performance of 30 different state-of-the-art techniques with respect to IR, class overlapping, and other issues. The findings from the investigation are detailed here.

3.1 What Makes Imbalanced Classification So Difficult

There are several factors contributing to the increased complexity of learning from imbalanced data. Some were identified in previous literature [8,9,48,66–68]. From the investigation conducted in this study, several other issues were noticed and highlighted below. Possible solutions to the problems are also discussed.

- **Rarity of the samples**: This is one of the primary causes of difficulty in learning the patterns from imbalanced data. When the number of minority-class samples available is limited, it naturally complicates the learning task. If obtaining more data is not an option, then OS is the only path to improve performance. However, this may increase the chance of overfitting [62].

- **Imbalance Ratio**: Class imbalance is another major contributing factor. When the data is skewed, classifiers naturally become biased towards the majority class. With the increase of IR, the situation worsens. The impact of IR on the performance of different techniques has been analyzed in this study. The findings are presented in the following section. Class distribution-based methods attempt to balance the data to reduce the effect of IR.

- **Class Overlapping**: It refers to the situation where data points from different classes occupy the same region in the feature space. Class overlapping occurs

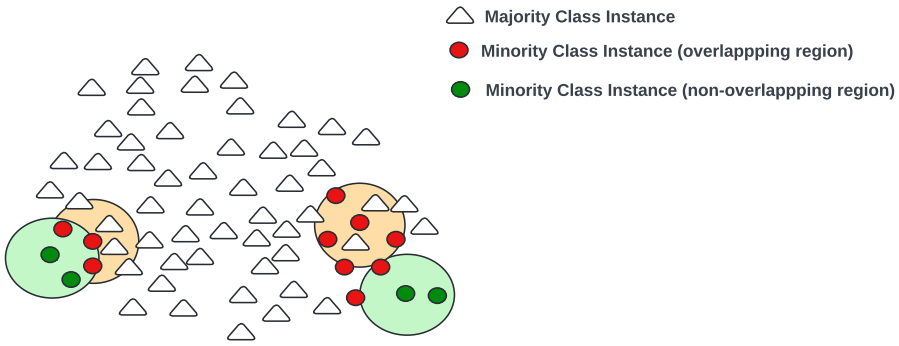

Figure 3.1: Class overlap.

when instances of different classes are not well-separated and share similar feature values in a dataset. This overlap makes it difficult for a classification algorithm to distinguish between the classes, leading to increased misclassification rates. Class overlapping is particularly problematic in imbalanced datasets, where the minority class has fewer samples, as it exacerbates the difficulty of accurately identifying instances of the minority class. This can significantly degrade the performance of ML models. Class overlap also has a strong correlation with class imbalance [69, 70].

Several authors have identified class overlapping as the primary cause behind the complicacy in imbalanced learning [3, 48]. If there is no overlap, classifiers such as SVM can easily distinguish positive cases from negative ones, even if the data is highly imbalanced. This factor intensifies the effects of the other two aforementioned factors. Necessary steps must be adopted to address this issue and obtain desirable performance from classifiers [61, 71].

The class overlapping scenario is illustrated in Fig. 3.1.

- **Noisy Samples**: Real-world datasets always contain some noisy instances. This can come from mislabeling some instances or other sources. Performing oversampling using such instances makes the dataspace more noisy. Besides, the synthetic samples generated by the OS techniques also introduce some noisy instances in the process. These samples can be difficult to identify. Moreover, most of the established techniques used in imbalanced learning do not take this into consideration, resulting in a loss of generalization and poor test performance.

- **Small Disjuncts**: Small disjuncts refer to subsets of the minority class that are sparsely represented in the feature space, often isolated from other instances of the same class. These small, isolated groups can be particularly challenging for

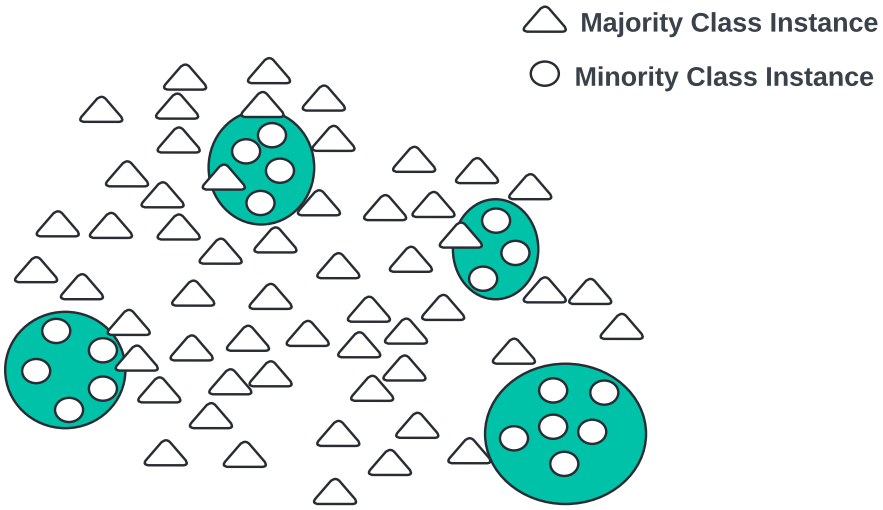

Figure 3.2: Small disjuncts in data.

classification algorithms to learn from, as they may not capture the overall structure of the minority class and can lead to increased misclassification [72]. They contribute to the overall difficulty of learning from imbalanced datasets, as the classifier may struggle to accurately identify and separate these small disjuncts from the majority class [73]. Smaller disjuncts are more susceptible to errors than larger ones, and most erroneous predictions arise from these smaller disjuncts [74].

The presence of small disjuncts in data is illustrated in Fig. 3.2.

Apart from these issues, the application of sampling techniques or CSL introduces new problems that limit the performance of the classifiers. These approaches also have certain limitations. Very few studies have been conducted to understand their behavior. On that account, a rigorous experimental study was conducted to analyze these issues. These are discussed in the following sections.

3.2 Experimental Framework

Extensive experiments were conducted on a wide range of imbalanced datasets with varying degrees of imbalance and overlapping scenarios. The performance of various techniques on these datasets was carefully analyzed. An overview of the entire experimental framework is illustrated in Fig. 3.3.

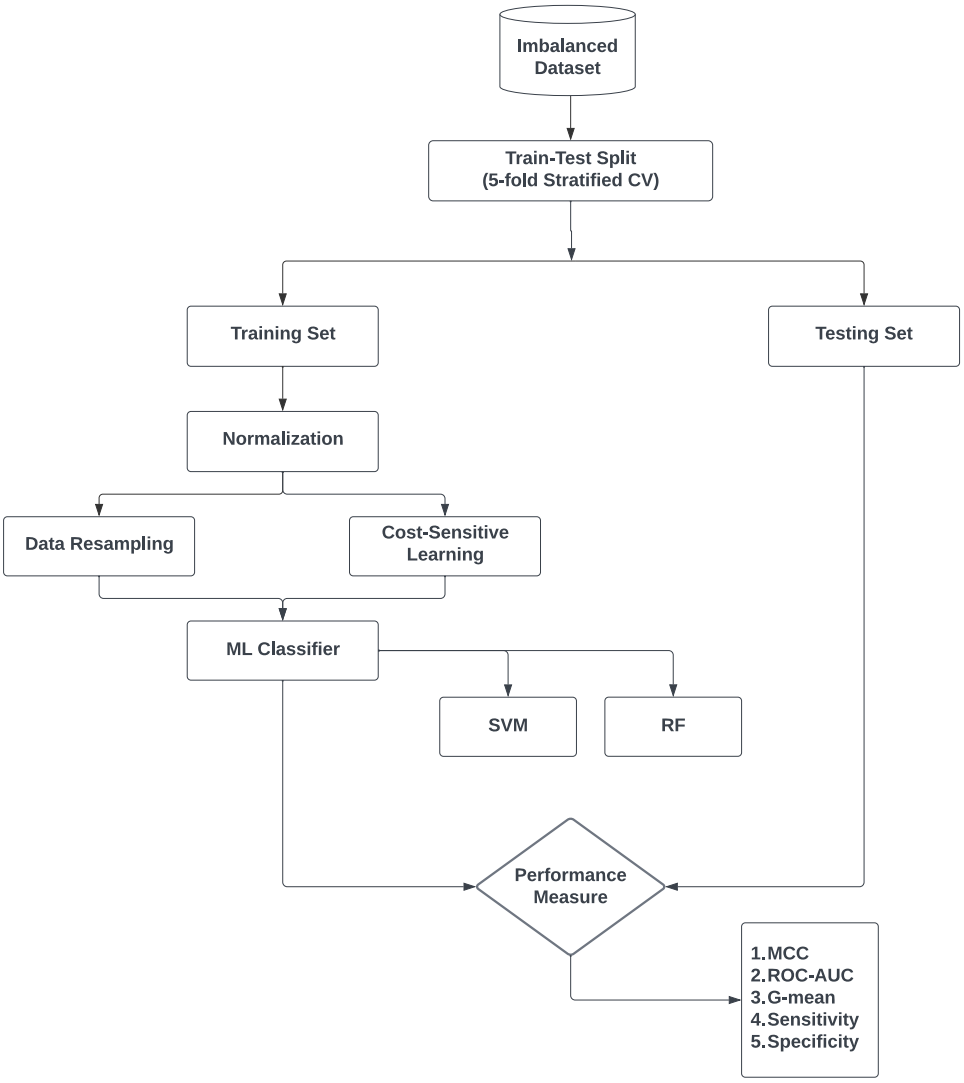

Figure 3.3: Outline of the experimental setup.

3.2.1 Datasets

A total of 84 real-world datasets were utilized for this experiment. The datasets were collected from various sources including UCI [75], and KEEL [76] data repositories. All the datasets utilized here are publicly available with no missing entries. All are binary classification scenarios. The IR of the datasets varied from 1.8 to 129. A summary of the datasets utilized in this study is provided in Table 3.1.

3.2.2 Methodologies

A total of 30 different methodologies were tested in this experiment. The most popular and recent methods were chosen for the experiment. This includes 13 OS techniques, 7 US techniques, 7 ensemble techniques, 2 hybrid sampling techniques, and cost-sensitive learning. The algorithms were chosen from all different categories to

Table 3.1: Summary of the datasets used in the experiment - I

Serial	dataset	# Samples	IR	Serial	dataset	# Samples	IR
1	glass1	213	1.8	43	glass4	214	15.38
2	wisconsin	683	1.86	44	ecoli4	336	15.75
3	pima	768	1.87	45	page-blocks-1-3_vs_4	472	15.82
4	glass0	213	2.09	46	abalone	731	16.4
5	yeast1	1483	2.46	47	dermatology-6	358	16.85
6	vehicle2	846	2.88	48	glass-0-1-6_vs_5	184	19.33
7	vehicle1	846	2.9	49	shuttle-6_vs_2-3	230	21.9
8	vehicle3	846	2.99	50	yeast-1-4-5-8_vs_7	693	22.07
9	vehicle0	845	3.27	51	flare-F	1066	23.77
10	new-thyroid1	215	5.11	52	car-good	1728	24.03
11	ecoli2	336	5.44	53	car-vgood	1728	25.57
12	glass6	214	6.34	54	kr-vs-k-zero-one_vs_draw	2901	26.88
13	yeast	1484	8.1	55	yeast4	1484	28.08
14	yeast3	1484	8.1	56	kr-vs-k-one_vs_fifteen	2244	28.13
15	ecoli3	336	8.57	57	winequality-red-4	1599	29.15
16	page-blocks0	5472	8.79	58	yeast128	947	30.53
17	ecoli-0-3-4_vs_5	200	8.95	59	yeast5	1484	32.7
18	ecoli-0-2-3-4_vs_5	202	9.05	60	abalone-3_vs_11	502	34.79
19	ecoli-0-6-7_vs_3-5	222	9.05	61	kr-vs-k-three_vs_eleven	2935	35.22
20	glass-0-1-5_vs_2	172	9.06	62	winequality-red-8_vs_6	656	35.39
21	yeast-2_vs_4	514	9.06	63	ecoli_013vs26	281	39
22	ecoli-0-4-6_vs_5	203	9.1	64	abalone-17_vs_7-8-9-10	2338	39.29
23	yeast-0-3-5-9_vs_7-8	506	9.1	65	yeast6	1483	41.37
24	glass-0-4_vs_5	92	9.11	66	abalone-21_vs_8	581	43.62
25	yeast-0-2-5-6_vs_3-7-8-9	1004	9.13	67	winequality-white-3_vs_7	900	43.95
26	yeast-0-2-5-7-9_vs_3-6-8	1004	9.13	68	winequality-red-8_vs_6-7	855	46.44
27	ecoli-0-2-6-7_vs_3-5	224	9.14	69	kddcup-land_vs_portsweep	1060	49.48
28	ecoli-0-3-4-6_vs_5	205	9.2	70	abalone-19_vs_10-11-12-13	1622	51.29
29	ecoli-0-3-4-7_vs_5-6	257	9.24	71	kr-vs-k-zero_vs_eight	1460	55.12
30	ecoli-0-6-7_vs_5	220	9.95	72	winequality_white	1481	58.24
31	vowel	988	9.98	73	winequality-white-3-9_vs_5	1484	58.24
32	glass-0-1-6_vs_2	192	10.24	74	poker-8-9_vs_6	1484	58.36
33	ecoli-0-1-4-7_vs_2-3-5-6	336	10.55	75	winequality-red-3_vs_5	691	68
34	glass-0-6_vs_5	108	10.89	76	abalone_20	1916	72.65
35	led7digit-0-2-4-5-6-7-8-9_vs_1	443	10.95	77	kddcup-buffer_overflow_vs_back	2233	73.4
36	glass-0-1-4-6_vs_2	205	11	78	kddcup-land_vs_satan	1609	79.45
37	glass2	214	11.53	79	kr-vs-k-zero_vs_fifteen	2193	80.19
38	ecoli-0-1-4-7_vs_5-6	332	12.24	80	poker-8-9_vs_5	2074	81.96
39	cleveland-0_vs_4	177	12.54	81	poker_86	1477	85.82
40	ecoli-0-1-4-6_vs_5	332	12.95	82	kddcup-rootkit-imap_vs_back	2225	100.09
41	shuttle-c0-vs-c4	1829	13.86	83	kddr_rookkit	2225	100.14
42	yeast-1_vs_7	459	14.27	84	abalone19	4174	129.41

obtain a good understanding of how different types of approaches fare in imbalanced classification tasks. Providing a detailed description of the algorithms is outside the scope of this manuscript. Detailed information on the algorithms is available in the original papers. The methodologies utilized in the experiment are listed in Fig. 3.4.

Two ML models were considered for the classification: Random Forest (RF) and SVM. Both of these classifiers were employed using the sklearn library.

3.2.3 Setup

Working with imbalanced data requires careful handling. To prevent data leakage, the data was initially split into training and testing sets. Only the training set was re-sampled, while the testing set remained untouched and was used solely for validation purposes. Data normalization was performed before sampling using MinMax scaling. No feature selection was applied. A stratified 5-fold cross-validation scheme was em-

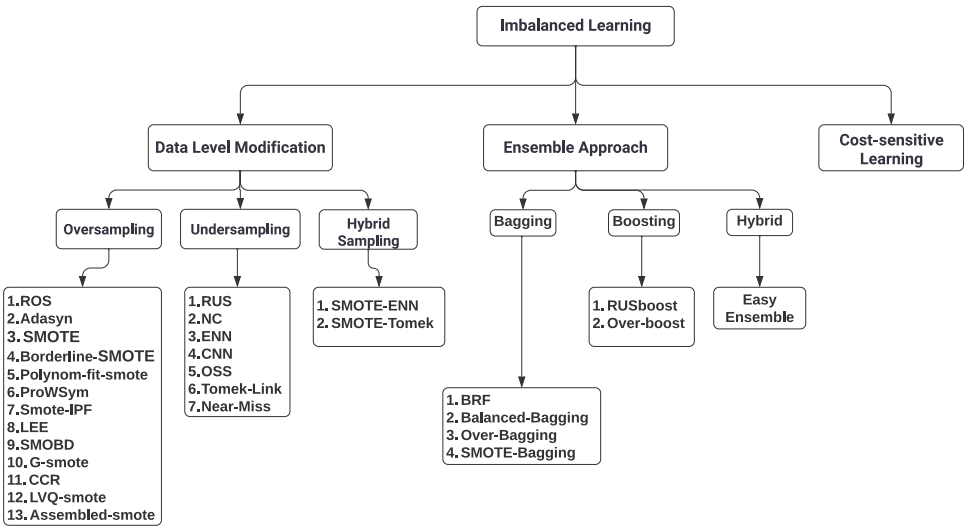

Figure 3.4: The methodologies utilized in experiment I.

ployed, and the average results from the five different testing folds were considered.

The sklearn library was utilized to implement the ML models. The imblearn and smote-variants libraries were utilized to implement the sampling techniques. The default parameter settings of the libraries were adopted. No hyperparameter tuning was performed. Five different measures were calculated to evaluate the performance: MCC, G-mean, ROC-AUC, sensitivity, and specificity.

For ease of discussion, the datasets were grouped into 3 categories based on their IR. Analysis and discussions have been provided accordingly.

- low imbalance: IR<10
- mid imbalance: IR=10-30
- high imbalance: IR>30

To assess the class overlap before and after resampling, we utilized the augmented R-value [71], an extension of the R-value [77] designed for imbalanced data. This extended version of the R-value provides a more nuanced assessment of how well classes are separated. It quantifies the degree of overlap between classes, which is crucial for understanding the separability of different classes in the feature space. R-values range from 0 (no overlap) to 1 (complete overlap).

3.3 Results and Discussion

In this section, the performance results obtained from the experiments have been presented. Due to the numerous techniques tested across a wide range of datasets, providing detailed performance measures for each algorithm on every individual dataset

is impractical for this article. These specific results can be found in the associated GitHub repository. The average of the results on all datasets has been presented in this manuscript. The MCC score is primarily considered for comparison due to its robustness. Other composite metrics show similar performance (provided in supplementary files).

The average MCC scores obtained using the SVM and RF classifiers as well as the ensemble methods are provided in Table 3.2, 3.3, and 3.4, respectively.

3.3.1 Performance analysis of classifiers with no sampling

With no sampling performed in the data, the performance from the classifiers is usually the lowest. However, there is a noticeable variation in different ranges of IR. For instance, the average MCC score obtained from the datasets with IR<10 is 59.90% for the SVM classifier. In comparison, the value is only 36.29% on datasets with IR>30, demonstrating a clear drop in performance.

The SVM classifier usually benefits more from sampling than the RF classifier. The average MCC score obtained from the RF classifier in low IR cases is 67.65%, almost 8% higher than the SVM classifier. The RF classifier constantly provides better performance than the SVM classifier in all imbalanced scenarios.

In low IR cases, the performance improvement from sampling is small. RF works pretty well without sampling in these low IR settings. However, as the IR increases, the difference in performance becomes apparent. Especially, in high IR scenarios, there is a major difference in performance indicating the importance of using some measures to address the class imbalance issue.

3.3.2 Performance analysis of the oversampling techniques

A total of 13 OS algorithms were evaluated in this study. Nearly all of these techniques significantly enhanced performance across various imbalanced scenarios. The performance of different SMOTE variants showed little variation. Among the OS techniques, LVQ-SMOTE, LEE, Polynom-fit-SMOTE, and SMOTE-IPF emerged as top performers. LEE achieved the highest average Matthews Correlation Coefficient (MCC) score of 69.6% for datasets with an imbalance ratio (IR) of less than 10, and it also performed well on highly imbalanced datasets. Polynom-fit-SMOTE recorded the highest average MCC scores of 59.6% for datasets with an IR between 10 and 30, and 49.3% for those with an IR greater than 30. Although there is a noticeable decline in performance with higher imbalances, it is important to note that the OS algorithms outperformed all other sampling techniques.

The decline in performance can be attributed to the high IR, which necessitates generating a substantial number of new samples to balance the dataset. Generating

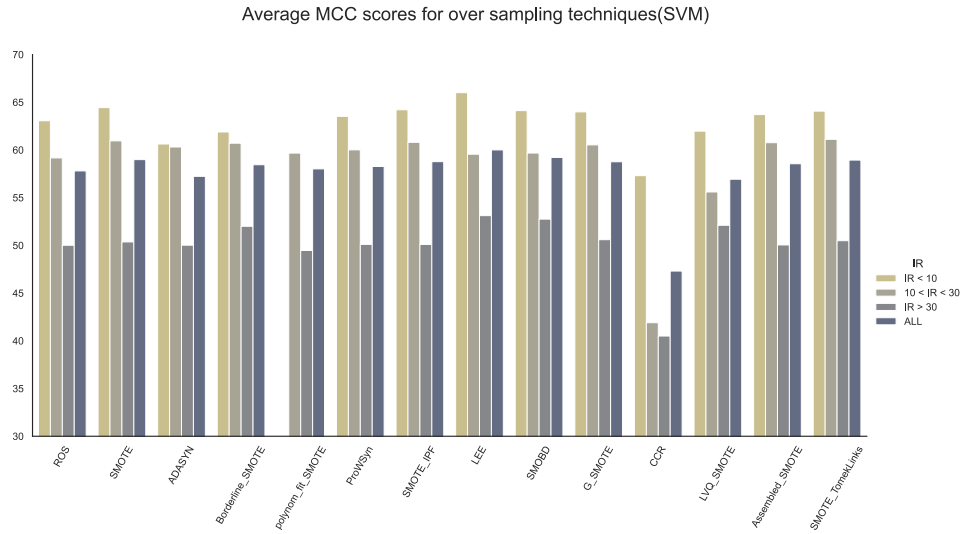

Figure 3.5: Performance comparison among the OS approaches for the SVM classifier.

many new samples from a limited number of examples can produce noisy samples that may not accurately represent the minority class, leading to a lower MCC score. Another limitation of these oversampling (OS) techniques is that most do not address the issue of overlapping. While increasing the number of minority-class samples improves performance, it also increases the overlapping with the majority-class instances.

Performance comparison among the OS approaches for different imbalanced scenarios is illustrated in Fig. 3.5.

3.3.3 Performance analysis of the undersampling techniques

A total of seven undersampling (US) approaches were tested in this study. While these techniques generally improved performance, some critical observations were noted. These techniques performed well with lower IR, but all showed a significant decline in performance as the IR increased. Among the under-samplers, the Neighborhood Cleaning Rule (NC) achieved the best results, with an MCC score of 68.25% for datasets with an IR of up to 10. However, NC's average MCC score dropped to 54.65% for datasets with an IR between 10 and 30, and it declined by an additional 12% for datasets with an IR greater than 30. Similar performance declines were observed with other US algorithms as well.

One key reason for this performance decline is that US techniques remove samples from the data to reduce overlapping and IR. Five of these US techniques focus on reducing overlapping, while only RUS and NearMiss aim to balance class distribution. When the IR is low, the overlapping-based techniques effectively manage to alleviate the scenario, resulting in high prediction performance. However, as the IR increases,

Table 3.2: Average MCC scores obtained using the SVM classifier (in percentage)

Method	$IR < 10$	$10 \leq IR \leq 30$	$IR \geq 30$	ALL
No sampling	59.9066	41.4498	36.2987	46.852
ROS	63.064	59.16717	50.01539	57.8051
ADASYN	60.6251	60.31106	50.02387	57.2376
SMOTE	64.4388	60.95442	50.36484	58.9897
BL-SMOTE	61.8858	60.70964	52.00332	58.4538
Polynom-fit-SMOTE	63.6602	59.674	49.46852	58.0188
ProWSyn	63.5236	60.011	50.10668	58.2696
SMOTE-IPF	64.2148	60.80276	50.10144	58.7759
SMOBD	64.126	59.67873	52.75457	59.2167
G-SMOTE	63.9946	60.53311	50.60008	58.7634
CCR	57.3101	41.89904	40.50158	47.3109
LEE	66.0069	59.55648	53.11975	60.0056
LVQ-SMOTE	61.9744	55.60123	52.10993	56.9351
Assembled-SMOTE	63.7104	60.76362	50.049	58.5561
RUS	59.8243	48.69019	38.97676	49.8989
NC	64.772	48.78341	38.87265	51.7723
ENN	63.3109	48.12443	39.68862	51.2668
CNN	59.6868	42.83827	34.56104	46.6603
OSS	61.7677	40.46503	36.35419	47.2696
Tomek-Link	61.5356	42.02024	37.03837	47.8765
Near-Miss	43.8894	30.95112	33.13978	36.538
SMOTE-ENN	62.6756	59.89414	49.89829	57.847
SMOTE-Tomek	64.3823	60.99517	50.36499	58.9809
CS-SVM	63.3358	59.27992	49.99665	57.9374

these methods fail to balance the data adequately. While they may reduce overlapping to some extent, the data remains significantly imbalanced, leading to a persistent bias towards the majority class.

On the other hand, RUS and NearMiss algorithms balance the distribution by removing samples, which leads to information loss. In highly imbalanced datasets, this can result in a significant loss of information, making the system unreliable for accurate predictions. Additionally, these US techniques do not increase the presence of minority-class samples in the data, causing the classifier to fail to correctly identify rare samples. This explains why the performance of US techniques is comparatively much lower than that of OS approaches.

Based on the experimental results, it can be concluded that while US strategies can effectively handle scenarios with lower class imbalance, they are entirely inappropriate

Table 3.3: Average MCC scores obtained using the RF classifier (in percentage)

Method	$IR < 10$	$10 \leq IR \leq 30$	$IR \geq 30$	ALL
No sampling	67.6556	50.6669	40.4831	53.9504
ROS	68.4453	59.0486	42.3317	57.4248
ADASYN	66.6317	57.5994	48.9715	58.3478
SMOTE	68.478	56.2975	46.883	57.9959
BL-SMOTE	66.8481	56.7566	45.6772	57.1459
Polynom-fit-SMOTE	68.7153	59.6327	49.348	59.886
ProWSyn	67.1571	56.5078	49.2264	58.2874
SMOTE-IPF	68.0929	57.5526	48.817	58.8396
SMOBD	68.183	55.9325	48.5586	58.2908
G- SMOTE	68.2207	58.4023	45.7059	58.1863
CCR	64.9599	49.9223	45.544	54.2674
LEE	69.5995	57.9977	46.6672	58.882
LVQ-SMOTE	66.157	58.0008	51.7078	59.1415
Assembled-SMOTE	67.8934	57.5967	46.9704	58.2045
RUS	62.8466	50.9698	39.5832	51.941
NC	68.246	54.6524	42.9226	56.1683
ENN	67.1091	53.5512	41.9704	55.0998
CNN	65.3988	52.4464	40.82	53.7512
OSS	67.2108	51.4952	39.4544	53.7195
Tomek-Link	68.4159	51.6779	40.8344	54.6616
Near-Miss	47.2432	32.8239	30.4928	37.5698
SMOTE-ENN	65.4524	58.721	49.8293	58.5148
SMOTE-Tomek	68.3949	56.1696	47.3205	58.0605
CS-RF	67.0366	52.2709	40.0523	53.7749

Table 3.4: Average MCC scores obtained using the ensemble methods (in percentage))

Method	$IR < 10$	$10 \leq IR \leq 30$	$IR \geq 30$	ALL
BRF	64.5724	54.061	40.5768	53.8633
Balanced-Bagging	62.6736	52.8011	40.6634	52.779
Over-Bagging	64.2763	52.0729	38.1633	52.385
SMOTE-Bagging	65.774	51.4461	41.5806	53.8749
RUSBoost	57.144	52.8859	39.2745	50.2768
Over-Boost	63.5572	58.5065	43.728	55.8358
Easy Ensemble	60.0863	53.036	36.3134	50.5205

and should be avoided when dealing with highly imbalanced data.

Performance comparison among the US approaches for different imbalanced scenarios is illustrated in Fig. 3.6.

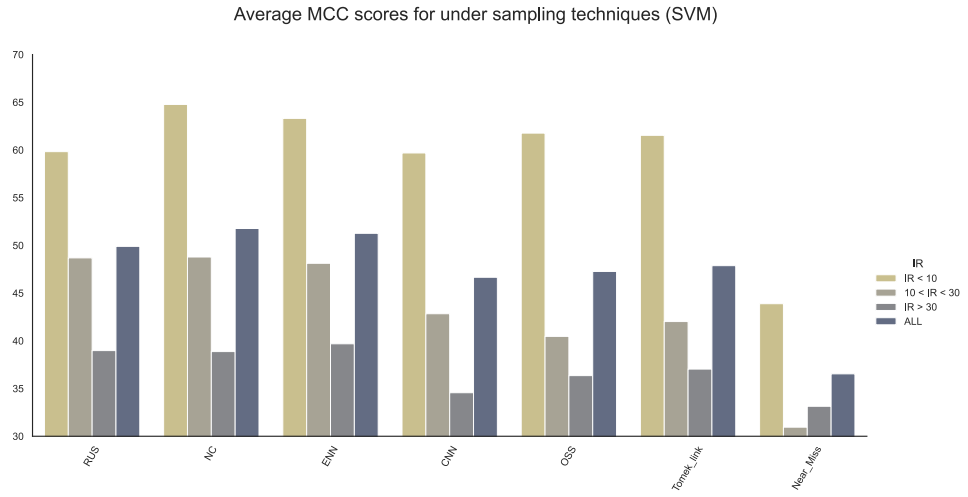

Figure 3.6: Performance comparison among the US approaches for the SVM classifier.

3.3.4 Performance analysis of the hybrid sampling techniques

The hybrid sampling approach combines both oversampling (OS) and undersampling (US) strategies to achieve a more balanced sampling method. In this study, two popular hybrid sampling approaches, SMOTE-ENN and SMOTE-Tomek, were tested. Both approaches yielded significantly better results than their undersampling counterparts but did not show much improvement compared to the SMOTE algorithm alone. Their performance was comparable to other OS techniques. For instance, SMOTE-ENN produced an MCC score of 57.85% for the SVM classifier. Compared to that, the US method ENN produced a much lower MCC score of 51.26%. SMOTE, however, produced a slightly higher score of 59%. The hybridization did not produce much improvement over SMOTE and its variants. For the RF classifier, resampling the data with SMOTE-ENN (MCC score = 58.5%) worked marginally better compared to SMOTE (MCC score = 58%).

Hybridization has promising potential, particularly in highly imbalanced scenarios, as evidenced by some recent applications [36] as well as our experiment. For highly imbalanced datasets, SMOTE-ENN (MCC score = 49.83%) performed much better than SMOTE (MCC score = 46.88%).

Despite its potential advantages, the exploration of hybrid methods in the existing literature remains relatively limited. Hybrid methods can simultaneously address both overlapping and IR problems, potentially leading to better results. Further investigation and empirical studies are recommended to comprehensively assess the efficacy and applicability of hybrid approaches across diverse datasets.

3.3.5 Performance analysis of the ensemble algorithms

Ensemble algorithms combine multiple weak learners to enhance prediction performance. In this research, we tested seven popular ensemble algorithms used in imbalanced learning, with Decision Tree as the base learner. SMOTE-Bagging and Over-Bagging are two OS-based bagging ensemble techniques, while RUSBoost and Over-Boost are boosting-based ensemble approaches. Although ensemble approaches aim to improve performance, this improvement was not realized in imbalanced cases. Most ensembles did not outperform the US techniques, and their performance was notably poor on highly imbalanced datasets. The results from the ensemble algorithms are shown in Table 3.

Among these algorithms, the OverBoost technique performed the best, achieving an MCC of 63.55% when the IR was less than 10. However, its performance dropped to 58.50% as the IR increased up to 30, and further degraded significantly for datasets with an IR greater than 30, scoring only 43.72%.

Bagging ensembles operate on bootstrapping, which creates random subsets of data. Weak learners are individually trained on these subsets, and predictions are aggregated later. However, this method does not address the imbalance issue, so the bootstrap subsets are resampled using RUS or SMOTE to achieve balance. Nevertheless, the original problems associated with these sampling techniques remain. Incorporating the RUS approach into an ensemble reduces the chance of information loss, resulting in slightly better performance from BRF or BB compared to RUS alone. However, since the RUS algorithm does not resolve the class overlapping issue, it persists in the ensemble approaches, leading to poorer performance compared to other US techniques.

The experimental results clearly indicate that directly preprocessing the entire dataset with SMOTE or its variants provides better results than combining them with ensemble learning frameworks, especially in highly imbalanced cases. To achieve better outcomes, the underlying sampling techniques must first address the issues of imbalanced learning.

Performance comparison among the ensemble approaches for different imbalanced scenarios is illustrated in Fig. 3.7.

3.3.6 Performance analysis of the cost-sensitive learning technique

CSL is an algorithm-level modification that increases the misclassification cost for minority samples. By imposing a higher penalty for misclassifying minority class samples, the algorithm is incentivized to minimize such errors to reduce the overall cost. In this study, different misclassification costs were assigned to each dataset, which was proportional to the IR of the specific dataset (default setting of the sklearn

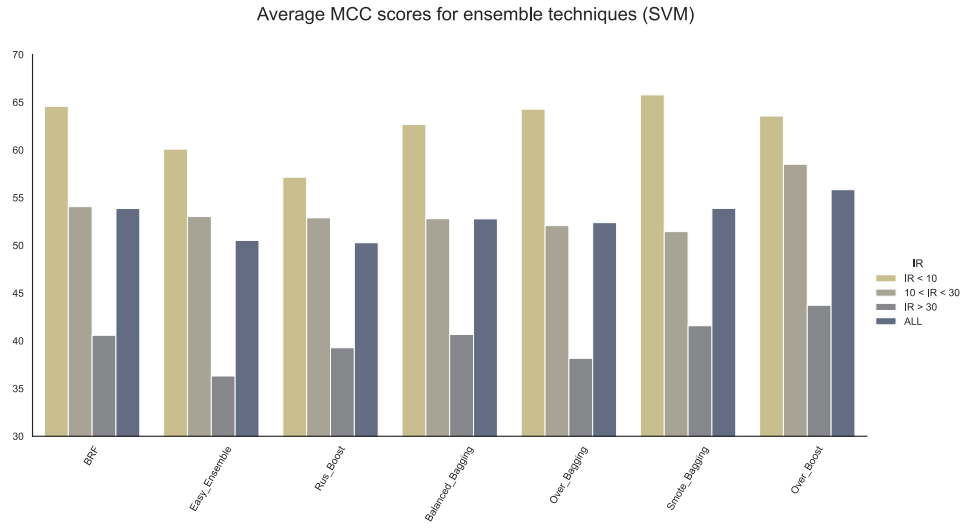

Figure 3.7: Performance comparison among the ensemble approaches.

library).

As can be observed from Table 3.2, the CS-SVM approach far outperforms the standard SVM classifier for all imbalanced cases. Especially in highly imbalanced cases, there was a significant boost in performance. The standard SVM classifier produced an MCC score of 36.3% in datasets with IR>30, whereas CS-SVM achieved an MCC score of 50%. CS-SVM also performed much better than US and ensemble methods. Its performance was comparable with the OS approaches. As for the RF classifier, this ensemble method is found to be less sensitive to CSL. Data resampling works much better compared to CSL in the case of the RF classifier.

3.3.7 Performance comparison of all the techniques

In this study, a total of 30 methods for handling imbalanced data were explored. Among the tested algorithms, Polynom-fit-SMOTE, a variant of SMOTE, achieved the highest average MCC score of 59.886% across all datasets. Most techniques performed well with less skewed data (IR < 10), but as the IR increased, the performance of all techniques declined. This decline was most pronounced in undersampling strategies. Ensemble approaches also struggled with larger imbalances and in highly imbalanced cases (IR > 30), the performance of both undersampling and ensemble approaches was significantly lower, making them unsuitable for such data. The performance fluctuation was the least among oversampling techniques.

Among all techniques, US approaches had the lowest average MCC scores. The best-performing undersampling algorithm, Neighborhood Cleaning (NC), scored only 51.77% (SVM), which is significantly lower than the scores achieved by OS approaches.

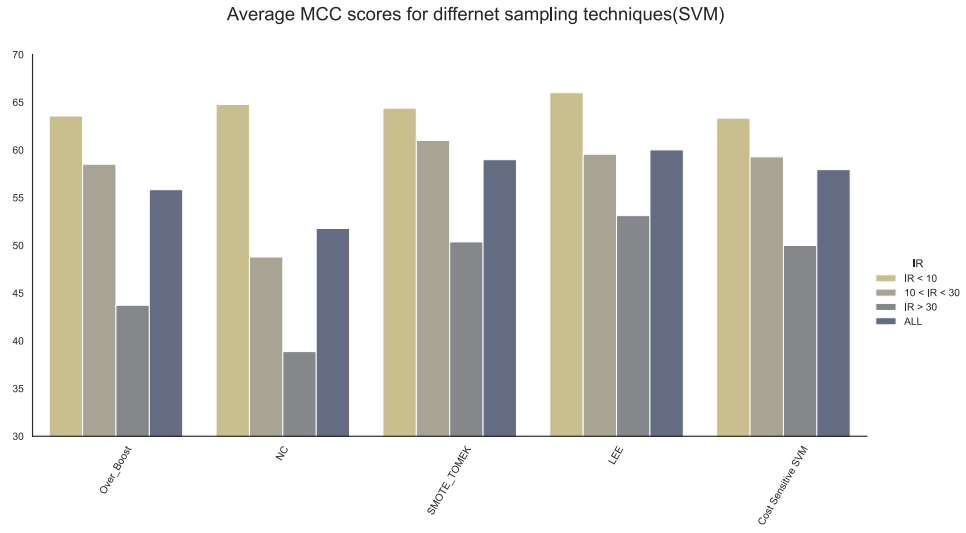

Figure 3.8: Performance comparison among different categories of approaches used in imbalanced learning.

Based on the results of all experiments, it can be concluded that SMOTE variants are much more effective in handling datasets with high imbalances compared to other imbalanced data handling approaches. The SVM classifier is more susceptible to class imbalance and benefits more from the application of sampling or CSL approaches.

Performance comparison among the different categories of approaches (best from each category) for different imbalanced scenarios is illustrated in Fig. 3.8.

3.4 Effect of Sampling Techniques on Class Overlapping

Class overlapping has been identified as one of the primary factors behind the performance drop in imbalanced data. Several investigations have been conducted by researchers on synthetic data to demonstrate its effect [3, 46, 70]. Santos et al. provided a unifying view of class imbalance with overlapping in their article [61]. The relation of class overlapping with the performance of the classifiers has been demonstrated in those articles as well. So, it has not been repeated in this manuscript. As the degree of overlap increases, it becomes progressively more challenging for classifiers to differentiate between opposing classes. Several methods have also been proposed to alleviate the issue [51, 78–80]. The traditional CSL approach does not consider class overlapping. No articles have been found addressing class overlapping while applying CSL.

Experiments were also conducted in this study to understand the effect of sampling techniques on class overlapping on real-world datasets. Two different measures: Augmented R-value [71] and degree overlap [4] were calculated to observe the level

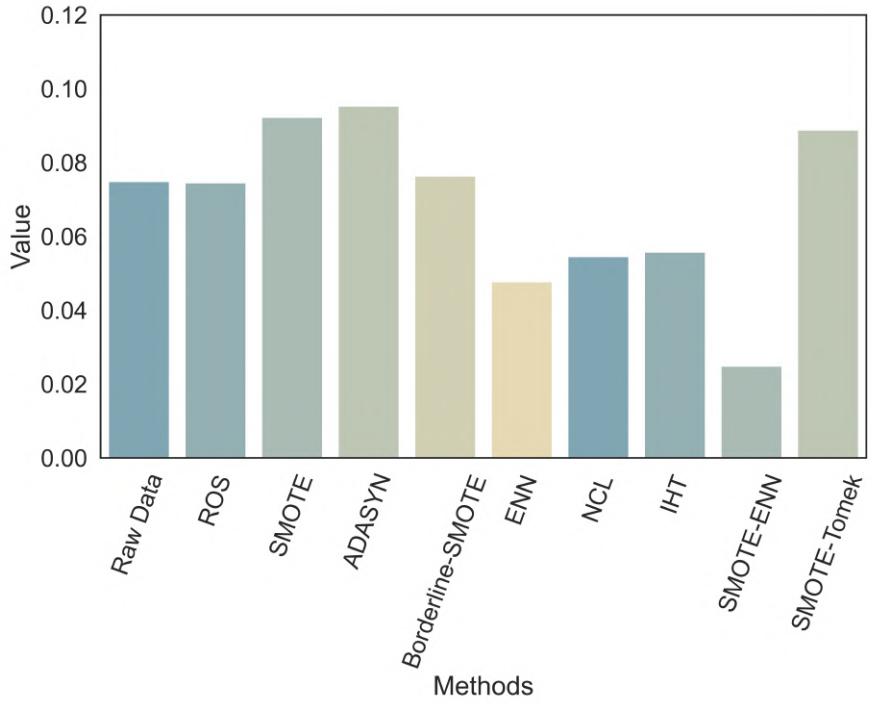

Figure 3.9: Effect of different sampling techniques on class overlapping.

of overlapping before and after sampling. Fig. 3.9 shows the effect on the datasets (average) based on augmented R-value for some selected sampling techniques. The detailed measures are provided in the supplementary files.

As can be observed from the figure, OS techniques such as SMOTE or ADASYN increase the overlapping while US approaches such as NCL or IHT reduce overlapping. The highest reduction has been observed in the SMOTE-ENN technique. Similar scenarios can be found in other sampling techniques as well. Although OS techniques raise the R-value, as has been presented in the previous section, OS techniques also produce the highest scores. This indicates and verifies that class overlapping is not the only contributing factor behind learning difficulties in imbalanced data. Other factors also play a key role in determining the performance of different techniques. All these factors together make learning difficult and therefore, all the factors need to be addressed to obtain desirable performance.

3.5 Limitations of Different Techniques Used in the Imbalanced Domain

In the previous sections, the performance of different techniques has been analyzed with respect to class imbalance and overlapping. Based on this investigation, a critical

review of the techniques has been provided here.

For this purpose, let us consider a scenario where the dataset has 10000 samples with an IR of 100. So, there are 100 instances belonging to the minority class, and the remaining 9900 samples belong to the majority class.

3.5.1 Undersampling

- RUS and Near-Miss are two class distribution-based US approaches. Such techniques remove the necessary number of samples from the majority class to balance the data. This can lead to unusual scenarios when the classes are significantly imbalanced. In the aforementioned scenario, balancing the distribution would require removing 9,800 majority-class samples, which is impractical and would result in a substantial loss of information. Although this approach would significantly improve the sensitivity score, it would also cause a decline in the specificity score. Even a small decrease in specificity indicates a large number of misclassifications, which is undesirable for any practical system.

- Other five techniques tested in this study (NCL, ENN, CNN, OSS, and IHT) are overlap-based approaches. These techniques try to reduce the class overlapping by carefully removing certain majority-class samples from the data. Different heuristics are adopted by the algorithms for selecting instances. These techniques do not balance the class distribution as the number of samples removed in the process is limited. As a result, they work well in low imbalances but fail in highly imbalanced datasets. If enough samples are not removed, the class imbalance persists and the classifier remains biased towards the majority class.

  For instance, in the above-mentioned dataset with 10000 samples, overlapping-based US techniques might eliminate 1400 samples. However, this would still leave 8,500 majority-class samples compared to only 100 minority-class samples. So, the data remains skewed even after sampling. Additional measures need to be taken to alleviate the scenario.

- The fact that the undersampling strategies do not enhance the presence of rare samples in the region is another significant problem with them. With few minority-class occurrences, it is challenging for the classifier to learn patterns. The uncommon samples may occasionally be seen by the classifier as noise in the data in extremely unbalanced situations. Raising the proportion of minority-class samples is essential for obtaining the high sensitivity that is frequently required in practical applications.

3.5.2 Oversampling

- Non-heuristic oversampling technique ROS just duplicates the existing samples to balance the class distribution. It does not add any new information. Duplicating samples this way usually causes overfitting and poor performance.

- SMOTE and its variants generate synthetic samples to balance the class distribution. Some of the variants such as BL-SMOTE and DBSMOTE are very careful as to where the samples are generated in the feature space (e.g., samples are generated along the decision boundary). Other variants generate samples everywhere in the feature space without any consideration. The question that has been raised is the authenticity of the synthetic samples that are generated. Some of the synthesized samples do not match the characteristics of the original minority class, increasing the chance of overfitting and poor performance while testing in real-world scenarios.

- Oversampling techniques like SMOTE and its variants aim to balance class distribution by generating the necessary number of minority-class samples. However, these algorithms generate samples arbitrarily without specific selection criteria for interpolation. A common issue with all these approaches is that in cases of high IR, an excessive number of samples must be generated to achieve balance.

  For example, in the scenario mentioned earlier, 9,800 samples would need to be generated from only 100 samples to attain balance. This can naturally lead to overfitting. To address this issue, the number of generated samples should not be excessively high.

- Another problem with OS techniques is the potential generation of noisy samples. Such noisy samples bias the classifiers resulting in the misclassification of majority-class instances. It is essential to carefully identify and remove these noisy samples from the data after sampling. Some SMOTE variants, such as LEE and SMOTE-IPF, incorporate this concept to address the issue.

- Another problem with applying OS techniques is that as new samples are generated, the dataset size is increased. This becomes problematic when dealing with large datasets. Oversampling can nearly double the size of the original dataset, significantly slowing down the training process and making it excessively time-consuming for big data applications.

3.5.3 Ensembles

- The aforementioned problems of the US techniques persist to some extent when these methods are integrated into an ensemble learning framework. The imbalance issue is not solved by forming an ensemble. However, because of bootstrapping, there is less information loss in the bagging process.

- Similarly, when OS methods are integrated into an ensemble learning framework, the original issues with the OS techniques persist. For instance, balancing each bootstrap subset with OS techniques will increase the size of each of them, significantly increasing the training time.

- While ensemble methods usually outperform traditional approaches, in the case of imbalanced data, this is not the case. The performance depends more on how well the data is resampled for training. Performance improvement was also found to be limited.

3.5.4 Cost Sensitive Learning

In traditional cost-sensitive approaches, the same cost value is typically assigned to all minority-class instances, which raises significant concerns. Not all minority-class instances present the same level of difficulty; those closer to the decision boundary are more likely to be misclassified than those farther away. It is crucial to penalize more difficult-to-learn instances more heavily than others, based on their proximity to the decision boundary. Otherwise, this can create some concerning issues while learning.

For example, in a dataset with an IR of 100, all minority-class instances would be penalized 100 times more than any majority-class instances (default sklearn implementation). This approach can lead to an unusual distortion of the decision boundary, resulting in a higher number of misclassifications of majority-class instances (reduced specificity score) during testing. Imposing unnecessarily high penalties on minority-class samples located far from the decision boundary biases predictions toward the minority class. This causes overfitting and loss of generalization. As such, these type of weighted classifiers tend to perform poorly on test data.

Chapter 4

UniSyn: A Unified Sampling Framework to Jointly Address Class Imbalance and Overlapping

Based on the investigations conducted on the sampling techniques and their identified limitations, a novel data resampling framework has been proposed in this thesis. The proposed algorithm aims to minimize the drawbacks of the established approaches and address all the data difficulty factors of imbalanced learning. The proposed methodology is presented in this chapter.

4.1 Overview

After examining the various methods currently employed to address class imbalance, we see the necessity for a novel approach that can simultaneously handle all data complexity issues. The established approaches fall short of addressing all those issues properly. It's challenging for a single sampling algorithm to meet all these requirements. Therefore, we propose a unified approach through hybridization to achieve the objectives. The aim is to develop a sampling framework that delivers well-generalized performance across a variety of imbalanced datasets. In the proposed approach, we address all data difficulty factors and the limitations of existing sampling techniques in multiple stages. This enables the algorithm to perform effectively in both low and high-imbalance situations, as well as in scenarios with small and large overlaps. Rigorous experiments have been conducted to evaluate the performance of the proposed approach. The methodology has been compared with other state-of-the-art techniques used in imbalanced learning. The proposed algorithm significantly outperformed all other approaches across all four composite metrics. Its consistent delivery of optimal results and exceptional performance on highly skewed datasets demonstrate the robustness of the proposed approach and its superiority over other sampling techniques.

4.2 Background

It has been noted that overlap-based techniques achieve similar performance in datasets with low imbalance. However, only reducing overlap is often insufficient for highly skewed data, as these techniques tend to underperform with greater imbalances. In situations where minority class samples are scarce, it is crucial to enhance their presence to achieve satisfactory performance. Otherwise, the classifier will exhibit very low sensitivity. If the data continues to be significantly imbalanced even after sampling, the overall classification performance will suffer. This is the main limitation of the overlap-based approaches found in our investigations.

Conversely, simply balancing the class distribution does not ensure enhanced performance. If overlap is not addressed during the process, the situation may not significantly improve. A classifier can perform well even with high imbalance levels if there is no overlap. However, most real-world datasets inherently have some degree of overlap, which is a major factor contributing to low classification accuracy. Distribution-based techniques aim to balance the class distribution, but this can lead to excessive resampling, particularly in highly imbalanced scenarios. Creating too many synthetic samples can cause overfitting and these samples may not accurately represent the minority class. This also increases class overlap, complicating the decision boundary. Additionally, undersampling requires removing a significant number of samples to achieve balance, resulting in information loss and a substantial decrease in specificity. These are the main limitations of the distribution-based approaches found in our investigations.

In our proposed methodology, we want to attain a balance between these two categories of approaches while simultaneously considering other data difficulty factors. Therefore, an undersampling approach is utilized to reduce overlapping and an oversampling technique is utilized to increase the presence of minority-class instances as well as balance the data. However, since these can create some other issues as discussed earlier, certain strategies have been adopted to avoid such scenarios.

Methods like SMOTE and its variants generate samples indiscriminately, without specific selection criteria for interpolation. Consequently, many of the synthesized samples do not enhance the classifier's performance and are essentially useless. This also raises the chance of producing samples that do not accurately represent the original class, leading to increased overfitting. Techniques such as Borderline-SMOTE are more careful in boosting the presence of minority class samples in challenging areas. However, all oversampling methods tend to create noisy samples, which need to be addressed.

While oversampling significantly improves the sensitivity score, it also increases

the number of misclassifications for majority class samples due to heightened overlap. Therefore, evaluating the performance of sampling techniques should include considering the misclassification rates for both classes to ensure a comprehensive assessment.

4.3 Proposed Methodology

A novel unified sampling framework has been proposed in this thesis. It has been outlined in this section. The method is divided into four stages. They are discussed below.

4.3.1 Oversampling

The presence of minority-class samples in the dataset is crucial for achieving good sensitivity. In imbalanced data, the number of positive instances is often limited, making it challenging to achieve high sensitivity without increasing their number. Thus, oversampling techniques are necessary to boost the number of minority-class samples.

However, oversampling can introduce issues. Traditional oversampling methods generate the required number of samples to balance the class distribution, often randomly across the dataspace, as seen in SMOTE and similar approaches. This can lead to overfitting since the generated instances may not accurately represent the minority class. Additionally, creating too many new samples increases overlap and introduces noisy instances into the data.

To effectively address these issues and prevent complications, we developed a customized version of the SMOTE algorithm for oversampling. Initially, we categorized the minority class samples into four types: Safe, Borderline, Rare, and Outlier. This categorization was proposed by Napierala et al. [81]. This process uses the nearest neighbor rule to categorize samples based on their respective neighborhoods, considering the five nearest neighbors during the experiment. The categorization is shown in Fig. 4.1.

- **Safe**: Safe samples have at most one neighbor from the opposite class.
- **Border**: Borderline samples have two or three neighbors from the opposite class.
- **Rare**: Rare samples have four neighbors from the opposite class.
- **Outlier**: Outliers are minority class samples entirely surrounded by opposite class instances.

Among these categories, safe samples are usually far from the decision boundary and have a low chance of being misclassified, making oversampling in this region

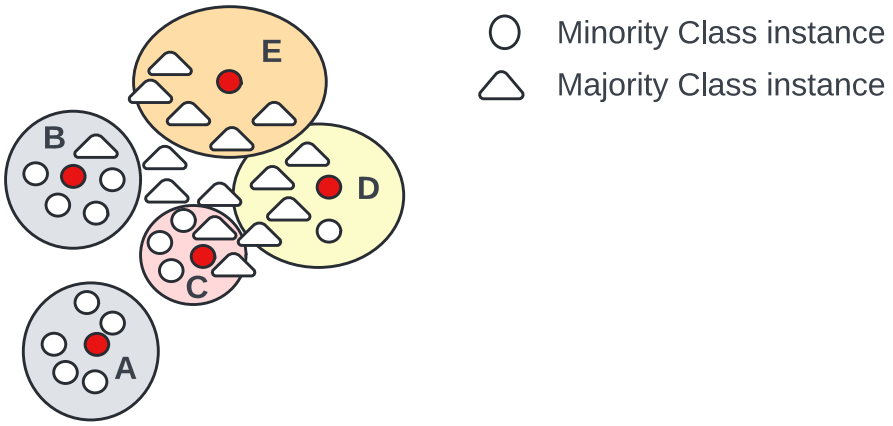

Figure 4.1: A representation of different categories of minority class instances: A and B are safe samples, C is a borderline sample, D represents a rare sample, and E is an outlier.

redundant. It is more effective to increase the number of minority-class samples in challenging areas. Therefore, we generated new synthetic samples around borderline and rare minority-class samples. The outliers are completely surrounded by instances from the opposite class. They are likely to be noisy instances. Using these instances to generate new instances can cause unusual synthetic samples that are most likely unrepresentative of the minority class. This can increase overfitting and poor generalization. Therefore, these instances are not considered during synthesis. This approach ensures that the generated samples are more likely to aid in classification. While generating samples near the decision boundary inevitably increases overlap, we limited the number of samples to reduce this effect and avoid overfitting.

For implementation, we first labeled all minority class samples according to the aforementioned categories. Then, the SMOTE algorithm was applied. The process involves randomly selecting a minority class sample from the training set. If the selected sample is labeled as safe, it is excluded from data synthesis. For other categories, we identify their k-nearest neighbors, typically using a k value of 5. A neighbor is randomly chosen from these k neighbors (excluding those in the safe category) and used to create a new sample through interpolation. This involves calculating the difference between the feature vectors of the two samples, multiplying it by a random number between 0 and 1, and selecting a random point along this line to form the new synthetic sample. The number of samples to be generated is controlled as a hyperparameter of the SMOTE approach.

We considered two aspects during oversampling. First, we did not aim to balance the dataset completely. Instead, we defined a parameter α to control the number of minority class samples generated. In our study, we set α to 0.5, meaning we generated

enough synthetic samples to reduce the IR by half. This is a tunable hyperparameter for optimization. Second, since oversampling inevitably creates some noisy samples, we applied a noise-filtering technique in the next stage to remove such samples from the data.

4.3.2 Data Cleaning

Applying oversampling techniques increases overlap with the opposite class and generates noisy samples, which negatively affects prediction performance. To mitigate this issue, we implemented a data-cleaning approach to eliminate noisy samples from the resampled dataset. We chose the Neighborhood Cleaning Rule (NCL) algorithm for this purpose. NCL identifies and removes noisy and ambiguous majority class samples, offering more thorough data cleaning compared to methods like the ENN algorithm or tomek-links. This also helps reduce overlap between the classes and decreases the imbalance ratio, though it does not completely balance the data. The NCL algorithm is usually used independently as an undersampling method. However, as has been demonstrated in Chapter 3, the performance from these types of approaches is not satisfactory in higher imbalances. They are not enough to tackle the class imbalance problem by themselves. Therefore, we utilize this approach as a part of our framework.

The NCL algorithm functions as follows: For each sample in the training set, its k-nearest neighbors (typically k=3) are identified. The original sample is then classified based on these neighbors. If the original sample belongs to the majority class and is misclassified by its neighbors, it is removed from the training set. Conversely, if the original sample belongs to the minority class and is misclassified by its neighbors, the majority class neighbors responsible for the misclassification are removed. This process identifies and eliminates noisy majority-class samples while preserving minority-class samples. The process is illustrated in Fig. 4.2.

This stage accomplishes three goals. First, the NCL approach clears the decision boundary of noisy samples. Second, it reduces class overlap, which helps with prediction accuracy. Third, it lowers the imbalance ratio, though the reduction is relatively

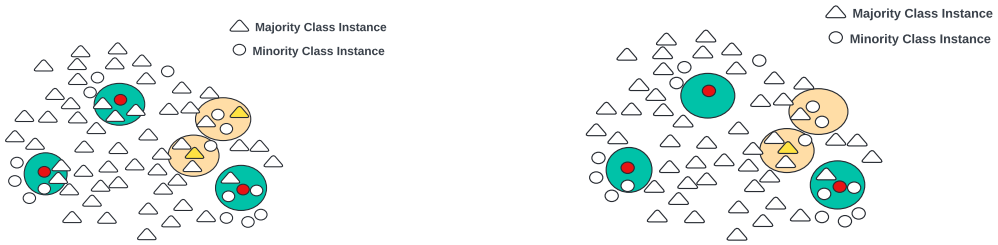

Figure 4.2: Majority class sample elimination using the NCL algorithm.

small (around 10-20% on average). Consequently, in highly skewed datasets, the class imbalance issue may persist. While reducing class overlap improves prediction, an imbalanced dataset can still bias the results. The NCL algorithm alone cannot completely avoid this bias, as evident from the lower performance measures reported in chapter 3 when it is applied independently.

4.3.3 Undersampling

A significant discrepancy in the number of samples between two classes negatively impacts prediction performance, making it essential to achieve a relatively balanced distribution. Although oversampling techniques were originally designed to balance class distribution by generating the necessary minority-class samples, this can lead to overfitting and other issues. To mitigate these problems, we only reduced the imbalance ratio (IR) by 50% during oversampling. The subsequent data-cleaning stage further reduces the IR to some extent, though the data may remain quite imbalanced.

To further lower the IR, we randomly removed some majority class samples from the data. Removing too many samples can lead to a loss of information, which can negatively affect performance. Given that the IR was already reduced in the previous two stages, only a limited number of instances needed to be removed. We introduced a hyperparameter, β, to control the number of samples removed in this process. In our study, β was set to 0.2, indicating a 20% reduction in IR through this approach. This value can be fine-tuned using grid search or similar methods. The aim of this step is not to balance the dataset entirely but to further reduce the IR, minimizing information loss, which is further addressed through bagging in the next stage. This step also reduces overlapping to some extent as a secondary benefit.

These three stages collectively form our proposed sampling framework. By working together, this methodology addresses all data difficulty factors while minimizing the shortcomings of individual techniques. Our proposed methodology is unique in its architecture, being the first approach to tackle all these issues simultaneously. We enhance it further by incorporating the sampling technique within an ensemble learning framework for improved generalization.

4.3.4 Ensemble learning

Ensemble learning is an ML approach that combines multiple learning algorithms to achieve better predictive performance than any single algorithm alone. This involves training a diverse set of learners on the same predictive task, allowing for greater variation in learning, better generalization, and lower variance. Standard ensemble methods include bagging, boosting, and stacking. These techniques often outperform traditional ML classifiers. However, creating an ensemble does not inherently solve the problem

of imbalanced classification, as both the ensemble and the base learners remain vulnerable to skewed class distributions. To address this, sampling techniques can be integrated into the ensemble learning framework by resampling the data used to train the base learners.

In this study, we combine the proposed sampling method with the bagging ensemble framework, specifically employing a customized bagging approach for imbalanced scenarios. The boosting framework can also be adapted similarly. Bagging helps mitigate information loss through bootstrapping, where many bootstrap subsets are generated from the original data, each used to train a specific weak learner. In one subset, certain samples might be missing or removed through undersampling, but in another subset, those samples might still be present and used to train a different classifier. Thus, using a large number (typically 100) of bootstrap subsets reduces the issue of information loss.

In bagging, bootstrap samples are generated from the original data through random sampling with replacement, producing multiple bootstrap samples that are potentially different from the original dataset. However, if the data is skewed, the bootstrap samples would be even more skewed after bootstrapping, and some subsets might lack minority-class samples entirely. To prevent this, we modify the original algorithm to include all minority-class samples in each bootstrap subset. Majority-class samples are then added with replacement to these subsets, ensuring that all bootstrap subsets have a similar imbalance ratio but vary in their constituent data points. Each subset is then independently resampled using the proposed sampling methodology. Consequently, each base learner is trained on a different subset of the data, and the predictions of the base learners are aggregated to obtain the final result. This approach helps the model achieve better generalization and robust performance. In our study, decision trees are used as the base learning algorithm, and the RF architecture is used to form the ensemble.

The proposed ensemble method is termed as 'iBRF: improved Balanced Random Forest Classifier'. This is a modified version of the original BRF classifier proposed by Chen et al. [37]. The original BRF classifier uses the simple RUS algorithm for sampling. However, as RUS removes too many samples to balance the data, it causes a loss of information and poor performance. Integrating the proposed sampling methodology alleviates the problem and enhances the prediction performance. This proposed iBRF algorithm far outperforms the original BRF classifier (performance measures are provided in subsequent sections). The architecture of the proposed iBRF classifier is presented in Fig. 4.3.

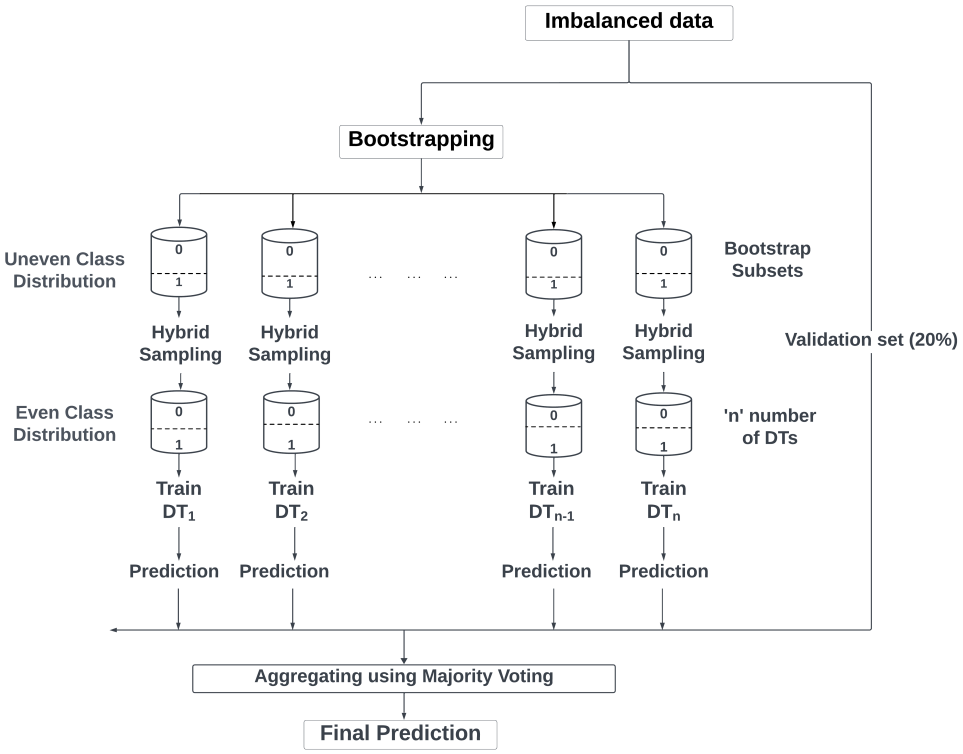

Figure 4.3: Architecture of the proposed iBRF classifier.

4.4 Experimental Framework

To validate the proposed model's preeminence, the algorithm has been tested on a range of imbalanced datasets. The performance is compared with other state-of-the-art techniques. Statistical significance tests have also been performed.

4.4.1 Datasets

The performance of the proposed approach was evaluated on 44 datasets. To assess the method's effectiveness across various imbalance scenarios, datasets with imbalance ratios ranging from 2 to 129 were chosen. These datasets were sourced from various repositories, including KEEL and UCI. Table 4.1 provides a summary of the datasets used in the experiments, all of which are real-world datasets for binary classification.

4.4.2 Experimental Setup

A similar experimental setup to experiment I (as described in chapter 3) was followed. For discussion related to the choice of performance measures or experimental setup, refer to previous chapters. The outline of the entire framework is presented in Fig. 4.4.

Table 4.1: Summary of the Datasets

Dataset Name	No. of Samples	No. of features	No. of Samples in the Minority Class	No. of Samples in the Majority Class	Imbalance Ratio
ionosphere	351	33	126	225	1.79
Wisconsin	683	10	239	444	1.86
pima	768	9	268	500	1.87
breast_cancer_wisconsin	699	10	241	458	1.9
cirrhosis	418	19	113	305	2.7
vehicle2	846	19	218	628	2.88
vehicle1	846	19	217	629	2.9
vehicle3	846	19	212	634	2.99
transfusion	748	5	178	570	3.2
spect_heart	267	23	55	212	3.85
new-thyroid1	215	6	35	180	5.14
ecoli2	336	8	52	284	5.46
mi_lethal_2	1547	102	232	1315	5.67
glass6	214	10	29	185	6.38
yeast	1484	9	163	1321	8.1
yeast3	1484	9	163	1321	8.1
ecoli3	336	8	35	301	8.6
yeast-2_vs_4	514	9	51	463	9.08
yeast-0-2-5-6_vs_3-7-8-9	1004	9	99	905	9.14
Satimage	4435	37	415	4020	9.69
vowel	988	14	90	898	9.98
led7digit-0-2-4-5-6-7-8-9_vs_1	443	8	37	406	10.97
glass2	214	10	17	197	11.59
ecoli-0-1-4-7_vs_5-6	332	7	25	307	12.28
cervical	753	31	53	700	13.21
glass4	214	10	13	201	15.46
ecoli4	336	8	20	316	15.8
page-blocks-1-3_vs_4	472	11	28	444	15.86
abalone	731	9	42	689	16.4
yeast-1-4-5-8_vs_7	693	9	30	663	22.1
yeast_ME2	1483	9	51	1432	28.08
yeast4	1484	9	51	1433	28.1
yeast128	947	9	30	917	30.57
winequality-red-8_vs_6	656	12	18	638	35.44
ecoli_013vs26	281	8	7	274	39.14
abalone-17_vs_7-8-9-10	2338	9	58	2280	39.31
winequality-white-3_vs_7	900	12	20	880	44
winequality-red-8_vs_6-7	855	12	18	837	46.5
abalone-19_vs_10-11-12-13	1622	9	32	1590	49.69
winequality_white	1481	12	25	1456	58.24
winequality-red-3_vs_5	691	12	10	681	68.1
abalone_20	1916	8	26	1890	72.69
kddcup-land_vs_satan	1609	39	20	1589	79.45
abalone19	4174	9	32	4142	129.44

4.4.3 Statistical Analysis

The statistical significance of the difference in performance between the approaches was determined using the Wilcoxon Signed Rank Test [82]. This is a non-parametric statistical test used to compare two related samples to assess whether their population mean ranks differ. It's an alternative to the paired Student's t-test when the data cannot be assumed to be normally distributed. It assumes that data are paired and come from the same population. Each pair is chosen randomly and independently. The null hypothesis is that the median of the differences between paired observations is zero. The significance level is considered to be 0.05 in this study. The p-values lower than that indicate a statistically significant difference between the two methods. The goal of statistical testing is to determine whether there is sufficient evidence to reject the null hypothesis in favor of the alternative hypothesis. The Wilcoxon Signed Rank Test is a robust tool for comparing related samples, especially when the normality assumption is questionable. It provides a way to assess differences in median ranks, making it valuable for a variety of experimental and observational studies.

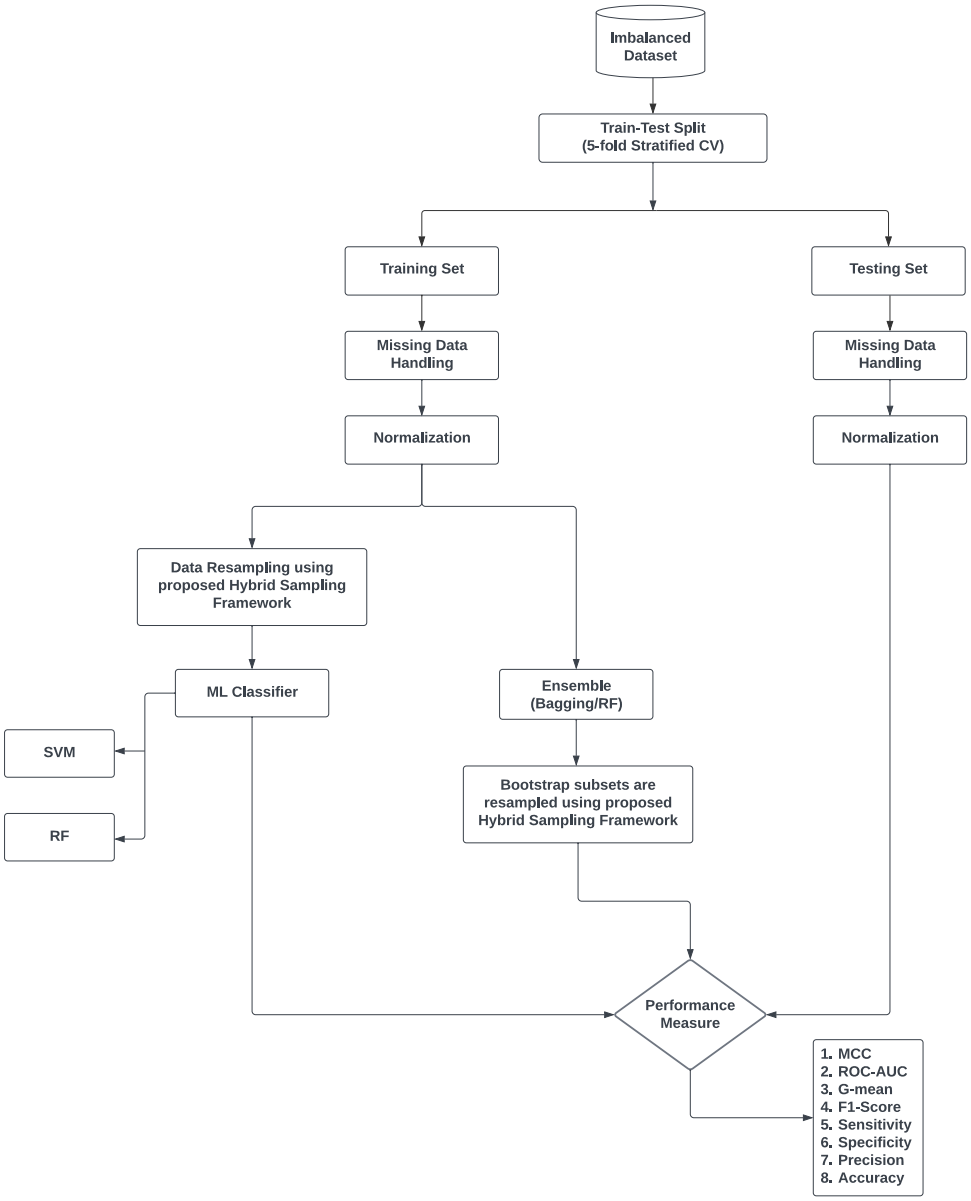

Figure 4.4: Outline of the experimental setup.

4.4.4 Performance Comparison

It is important to compare the performance of a method with other state-of-the-art techniques to determine where the method stands. As such, the performance of our proposed approach was compared with 28 other popular methods used for imbalanced learning. The methods are chosen based on their popularity and recency. Some of the best-performing SMOTE-variants as identified by Kovacs in their article [83] are also included for comparison. The algorithms are implemented using the imblearn library [84] and the smote-variants library [17] with default parameter settings.

The approaches used for performance comparison are listed below.

- Oversampling
  - Random Oversampling (ROS)
  - Synthetic Minority Oversampling Technique (SMOTE) [14]
  - Adaptive SMOTE (ADASYN) [16]
  - Borderline SMOTE (BL-SMOTE) [18]
  - Polynom-Fit-SMOTE [55]
  - Learning Vector Quantization SMOTE (LVQ-SMOTE) [85]
  - Geometric SMOTE (G-SMOTE) [25]
  - SMOBD [41]
  - LEE [54]
  - SMOTE-IPF [23]
- Undersampling
  - Random Undersampling (RUS)
  - Tomek-Link
  - Edited Nearest Neighbor (ENN)
  - Condensed Nearest Neighbor (CNN) [30]
  - Neighborhood Cleaning (NC) [49]
  - Instance Hardness Threshold (IHT) [32]
- Hybrid Sampling
  - SMOTE-ENN [33]
  - SMOTE-Tomek [33]
  - ROS+RUS
  - Random Balance [34]
- Ensemble Methods
  - Over-Bagging
  - SMOTE-Bagging [58]
  - Balanced-Bagging [86]
  - Over-Boosting
  - RUSBoost [60]

- Easy Ensemble [38]
- Balanced Random Forest (BRF) [37]

- Cost-Sensitive Learning [5]

4.5 Results and Discussion

The performance measures obtained from the proposed approach, its comparison with other sampling techniques, and the results from the statistical significance tests are provided in this section. The average results are reported in Table 4.2 and 4.3 for the SVM and RF classifiers, respectively. Detailed results on each individual dataset are provided in supplementary files.

Table 4.2: Performance of different approaches using SVM as the base classifier (in percentage)

Methods	G-MEAN	MCC	ROC	F1-Score	Sensitivity	Specificity	Accuracy
NO SAMPLING	42.997	37.515	66.952	38.739	35.969	**97.935**	**92.702**
ROS	75.267	48.191	79.458	50.831	73.452	85.464	84.877
SMOTE	74.787	48.21	78.961	50.9	71.581	86.341	85.595
ADASYN	74.377	46.9	78.955	49.744	73.096	84.814	84.539
BLSMOTE	72.352	48.669	78.217	51.956	68.737	87.698	87.11
POLYNYM-FIT-SMOTE	72.281	47.192	77.982	50.095	68.418	87.545	86.384
LVQ-SMOTE	73.544	46.673	78.328	49.782	70.102	86.553	85.605
G-SMOTE	74.099	48.397	78.811	51.141	70.958	86.665	85.82
SMOBD	75.587	49.145	79.455	51.517	72.83	86.081	85.203
LEE	68.061	49.27	76.37	52.527	60.961	91.78	89.997
SMOTE-IPF	73.887	47.994	78.766	50.753	70.889	86.643	85.677
RUS	76.475	44.836	79.008	47.616	80.304	77.712	78.038
TOMEK-LINK	45.302	38.617	67.867	40.786	38.83	96.904	92.582
ENN	51.249	41.666	70.596	44.589	47.377	93.815	91.629
NC	51.484	42.733	70.72	45.433	47.028	94.412	92.067
CNN	49.905	37.983	69.623	41.628	49.756	89.49	88.285
IHT	63.878	44.972	74.841	48.631	64.578	85.104	85.679
SMOTE-ENN	75.3	46.615	79.088	49.412	75.755	82.42	82.609
SMOTE-TOMEK	74.617	48.112	79.045	50.764	71.787	86.303	85.574
ROS + RUS	71.032	49.263	77.606	51.858	63.616	92.067	89.306
Random Balance	70.829	49.096	77.428	51.846	62.863	92.442	89.602
Proposed	**79.515**	**49.748**	**81.837**	**52.579**	**81.03**	82.642	82.936

4.5.1 Performance Analysis of the Proposed Unified Sampling Framework

As can be observed from the tables, for both SVM and RF classifiers, the proposed approach outperforms all the other approaches in terms of all four composite metrics. The table shows the average results on 44 imbalanced datasets. On average, there is a significant improvement in performance from the proposed approach. In individual scenarios also, the proposed approach usually outperforms the other approaches.

Table 4.3: Performance of different approaches using RF as the base classifier (in percentage)

Methods	G-MEAN	MCC	ROC	F1-Score	Sensitivity	Specificity	Accuracy
NO SAMPLING	51.47	44.04	69.41	45.24	41.68	**97.14**	**92.89**
ROS	57.662	46.37	71.726	49.019	47.932	95.521	92.271
SMOTE	63.459	47.268	74.005	51.065	54.138	93.873	91.201
ADASYN	62.836	46.433	73.716	50.294	53.856	93.577	91.025
BLSMOTE	60.228	45.999	72.622	49.725	50.748	94.496	91.671
POLYNYM-FIT-SMOTE	63.076	47.852	74.362	51.347	54.659	94.065	91.573
LVQ-SMOTE	68.313	49.47	76.301	52.681	59.553	93.049	90.537
G-SMOTE	60.436	46.959	72.922	50.128	51.082	94.761	91.791
SMOBD	66.296	48.231	75.16	51.905	57.352	92.967	90.517
LEE	58.109	47.284	72.592	49.918	49.532	95.651	92.528
SMOTE-IPF	63.581	47.35	74.123	51.118	54.426	93.82	91.124
RUS	77.373	44.474	79.532	47.318	**79.701**	79.362	79.448
TOMEK-LINK	53.391	45.032	70.14	46.793	43.981	96.298	92.647
ENN	58.818	46.961	72.699	49.8	52.247	93.15	91.418
NC	57.956	46.859	72.548	49.615	51.535	93.562	91.732
CNN	61.155	45.235	73.192	49.104	54.668	91.715	90.105
IHT	71.663	44.842	77.65	48.925	74.156	81.145	82.709
SMOTE-ENN	71.001	48.613	76.86	52.486	64.274	89.445	88.439
SMOTE-TOMEK	63.214	46.836	73.728	50.52	53.742	93.714	91.018
ROS + RUS	55.529	45.39	71.655	47.627	46.033	96.061	92.482
RANDOM BALANCE	62.959	48.759	74.548	52.045	52.746	95.13	91.98
Proposed	**80.08**	**52.324**	**82.515**	**54.805**	79.597	85.549	85.611

The proposed approach produced the highest MCC score of 52.314% and 49.748% for the RF and SVM classifiers, respectively. While the sensitivity score is also the highest from the proposed approach in the case of the SVM classifier, the RUS algorithm produced the highest sensitivity (79.701%) for the RF classifier. The sensitivity score from the proposed approach is also almost equal (79.597%). The classifiers trained on the unsampled data produced the highest specificity scores which indicates a clear bias from these traditional algorithms towards the majority class. Applying sampling techniques reduces the specificity score but improves the sensitivity significantly.

As for the F1-score, it is the harmonic mean of sensitivity and precision. Both of these metrics are class-specific and related to the performance on the minority class instances. As a result, a high F1 score is difficult to achieve as both of its constituents heavily depend on the minority class. That is why the average F1 score achieved is comparatively lower, similar to the MCC score. To further illustrate this, let's look at the following scenario.

- Total no. of instances = 100
- No. of minority class instances = 5
- No. of majority class instances = 95
- TP = 2

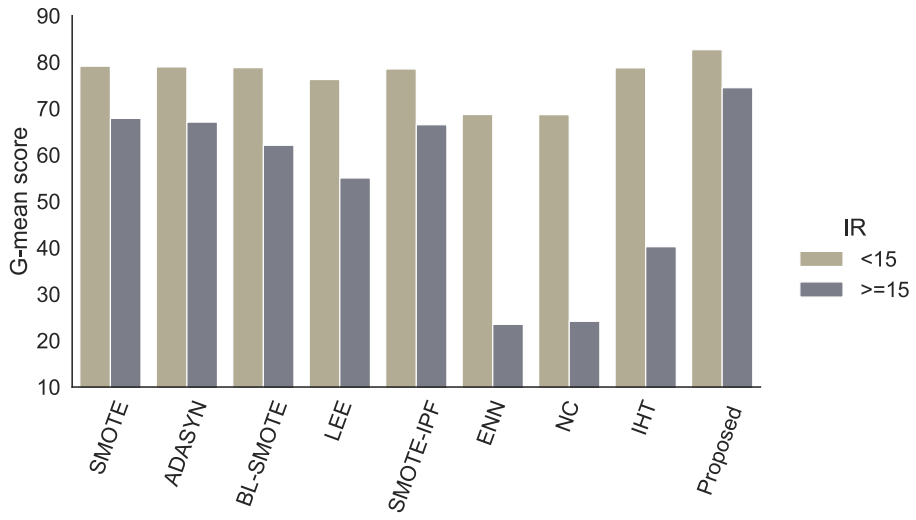

Figure 4.5: Difference in average performance based on the IR value for the SVM Classifier.

- TN = 90
- FP = 5
- FN = 3

This is a typical scenario. The performance measures obtained from such a case are as follows:

- Accuracy = 92% (misleading)
- Sensitivity = 40% (poor performance on the minority class)
- Specificity = 94.74% (indicating bias towards majority class)
- G-mean = 61,55%
- Precision = 28.57%
- F1 score = 33.33%

Overall, the proposed approach provides better performance in different imbalanced scenarios than other state-of-the-art sampling approaches. The variation in average performance across different sampling approaches, depending on the IR value, is depicted in Fig. 4.5. Detailed discussion is provided in the following sections.

4.5.2 Performance Comparison of the Proposed Approach with Undersampling Techniques

As shown in the tables, the performance improvement from applying undersampling techniques is minimal. Particularly in highly imbalanced datasets, these undersampling algorithms performed poorly, as evidenced by the dataset-specific results in the supplementary files. In several instances, both the sensitivity and G-mean scores are 0, indicating that the classifier remained biased towards the majority class despite the sampling. This is undesirable and suggests that undersampling techniques are ineffective in highly imbalanced scenarios.

It is evident from the tables that the RUS algorithm exhibits an unusually high G-mean score compared to other undersampling methods. This anomaly can be explained by examining the sensitivity and specificity scores. RUS achieves a relatively high sensitivity score at the expense of the lowest specificity score. As a class distribution-based sampling method, RUS removes a significant number of majority class samples to achieve balance. In cases of high imbalance, this results in the removal of too many samples, causing the bias to shift from the majority class to the minority class. Consequently, RUS yields a high sensitivity but low specificity score. While high sensitivity is generally desirable, low specificity indicates a high rate of majority-class misclassifications. This is not suitable for a reliable decision support system, where balanced predictions for both classes are preferable. Other metrics, such as MCC and ROC-AUC, reflect this issue.

The other five undersampling techniques included in the comparison are overlap-based, aiming to reduce class overlap by strategically removing certain majority class samples. However, these techniques do not balance the class distribution, resulting in the data remaining skewed and the bias towards the majority class persisting. This is reflected in the performance metrics of these methods, which show the poorest sensitivity but the highest specificity among all sampling approaches. Consequently, their G-mean and ROC-AUC scores are also the lowest.

Regarding MCC scores, IHT and NC are the top-performing undersampling approaches. The NC algorithm achieves an MCC score of 42.73% and a ROC-AUC of 0.7 for the SVM classifier. In contrast, our proposed sampling technique performed significantly better, with an MCC score of 49.75% and a ROC-AUC score of 0.82. Similarly, for the RF classifier, our proposed method outperformed the other undersampling approaches by a considerable margin.

4.5.3 Performance Comparison of the Proposed Approach with Oversampling Techniques

This study evaluated ten different oversampling techniques for comparison. These techniques generally perform well, often surpassing the undersampling methods. Oversampling techniques are particularly effective in achieving good sensitivity, even in cases of high imbalance where undersampling techniques tend to fail. This success is attributed to the increased presence of minority-class instances, which is essential for good performance. Among the oversampling techniques, LEE and LVQ-SMOTE produced the highest MCC scores for the SVM and RF classifiers, respectively.

As seen in the tables, the proposed sampling approach significantly enhances performance. For the RF classifier, SMOTE achieved an MCC score of 0.47 and a G-mean score of 0.63, with other SMOTE variants producing similar results. In contrast, our proposed approach achieved an MCC score of 0.53 and a G-mean score of 0.8, markedly higher than the oversampling techniques. It also outperformed the best SMOTE variants; for instance, LVQ-SMOTE achieved a G-mean score of 0.735 with the SVM classifier, compared to 0.791 from our proposed approach, demonstrating the framework's ability to improve performance.

In terms of sensitivity, the proposed framework showed unmatched performance. Undersampling techniques yielded the lowest sensitivity scores, while oversampling techniques improved sensitivity by generating new synthetic samples, with scores around 70% for the SVM classifier. ADASYN achieved the highest sensitivity of 73% among these methods. Our proposed approach, however, achieved a sensitivity score of 81%, significantly higher than the other techniques. This improvement is due to the generation of more representative minority-class instances and the removal of noisy and overlapping samples, which helped the classifier better distinguish instances from different classes.

4.5.4 Performance Comparison of the Proposed Approach with Hybrid Sampling Techniques

Hybridization of oversampling and undersampling techniques has not been extensively explored. Previous studies typically aimed to balance class distribution using combinations like ROS and RUS or SMOTE and RUS [34]. However, these approaches do not address issues of overlapping or noisy samples. Simply combining two techniques fails to resolve the core challenges of imbalanced learning, as evidenced by the low G-mean, MCC, and ROC-AUC scores from the ROS+RUS and Random Balance (SMOTE+RUS) hybrid methods. These algorithms only achieved G-mean scores of 55.52% and 62.96%, respectively, for the RF classifier, whereas our proposed algo-

rithm significantly outperformed them with a G-mean score of 80.08

Other hybrid methods, such as SMOTE-Tomek and SMOTE-ENN [33], combine SMOTE with Tomek-links and ENN undersampling methods. Tomek-link undersampling removes majority-class instances from the identified Tomek-links, while ENN removes borderline instances. The issue with these approaches is that they remove too few samples, leaving the data quite skewed and necessitating the generation of many samples by SMOTE. Additionally, they do not address the noisy and overlapping samples created by SMOTE, leading to other complications discussed in the discussion section.

Among these hybrid methods, Random Balance achieved the highest MCC score of 0.48, and SMOTE-ENN reached the highest ROC-AUC score of 0.76 for the RF classifier. Although these hybrid approaches perform better than pure oversampling or undersampling methods, our proposed method offers a significant improvement (around 4-5%) over these hybrid techniques. This indicates that our strategically designed sampling framework is more effective at addressing the imbalanced classification problem.

4.5.5 Performance Comparison of the Proposed Approach with Cost-Sensitive Learning

The average performance metrics from the CS approaches are summarized in Table 4.4, with detailed results for each dataset available in the supplementary files. The results from both SVM and RF classifiers demonstrate that our proposed approach consistently outperforms the cost-sensitive method. Although the CS-SVM shows better performance compared to the CS-RF approach, our sampling framework achieves significantly higher scores across various metrics, including G-mean, MCC, ROC-AUC, F1-Score, and sensitivity. The RF classifier, in particular, is found to be less responsive to the CS approach. The CS-RF delivers the highest specificity score but the lowest sensitivity, indicating a strong bias towards the majority class. This can be explained by the RF's use of the bagging process to generate bootstrap subsets of the data. When dealing with imbalanced data, the original bootstrapping process exacerbates class imbalance within these subsets, sometimes resulting in the complete absence of minority-class instances in highly imbalanced scenarios. To address this, we proposed a modified version of the bagging process in this study, ensuring an adequate presence of minority-class samples in all bootstrap subsets.

Table 4.4: Performance comparison of the proposed approach with cost-sensitive learning (in percentage)

	G-mean	MCC	ROC-AUC	F1-Score	Sensitivity	Specificity	Accuracy
CS-SVM	75.58	47.68	79.21	50.39	73.61	84.81	84.34
Proposed Sampling (with SVM)	79.515	49.75	81.84	52.58	**81.03**	82.64	82.94
CS-RF	50.18	42.94	69.25	44.51	41.33	**97.18**	**92.92**
Proposed Sampling (with RF)	**80.08**	**52.324**	**82.515**	**54.81**	79.6	85.55	85.61

4.5.6 Performance of the Proposed iBRF algorithm and its Comparison With Other Ensemble Techniques

The performance measures from the ensemble methods are provided in Table 4.5. Among the ensemble techniques evaluated in this study, the Over-Boost method achieved the highest MCC score at 0.47, while BRF attained the highest ROC-AUC score of 0.81. Our proposed hybrid ensemble method surpasses these techniques, achieving an MCC score of 0.53 and a ROC-AUC score of 0.823. Additionally, the hybrid ensemble outperforms our baseline sampling approach used for resampling the bootstrap subsets. This is expected, as integrating the baseline sampling with the ensemble method enhances generalization and results in more robust performance.

Table 4.5: Performance comparison of the proposed iBRF algorithm with other ensemble techniques (in percentage)

Methods	MCC	G-MEAN	ROC	Sensitivity	Specificity	Accuracy	F1-Score
Over-Bagging	42.325	51.665	69.128	41.887	**96.370**	**92.117**	44.593
SMOTE-Bagging	44.463	56.839	71.123	46.885	95.361	91.757	47.720
Balanced-Bagging	45.823	75.662	78.169	71.092	85.245	83.909	49.237
Over-Boosting	47.590	69.911	76.193	62.021	90.366	88.606	51.650
RUSBoost	41.037	68.597	74.313	62.245	86.382	84.640	46.212
Easy Ensemble	44.325	78.674	80.425	**82.604**	78.246	78.687	47.105
BRF	47.031	79.244	81.044	81.776	80.312	80.514	49.095
iBRF (proposed)	**53.042**	**79.923**	**82.260**	78.931	85.589	85.880	**55.002**

4.5.7 Statistical Significance Test

The results from the Wilcoxon signed rank tests are presented in Table 4.6. As shown in Table 4.6, the performance improvements in terms of G-mean, MCC, and ROC-AUC scores achieved by our proposed approach are statistically significant compared to all other methods. For other metrics, the performance improvements are not as significant in only a few instances. A similar trend is noted for the SVM classifier, with detailed Wilcoxon test results provided in the supplementary files. We also conducted a statistical significance test for our proposed hybrid ensemble method against other ensemble techniques, with p-values reported in Table 4.7. These results demonstrate that the performance differences between our proposed method and other ensemble methods are statistically significant for almost all metrics.

Table 4.6: p-values of the Wilcoxon Signed Rank Tests for the proposed algorithm compared to other sampling techniques (RF as the base classifier)

Methods	G-MEAN	MCC	ROC	F1-Score	Sensitivity	Specificity	Accuracy
NO SAMPLING	1.21E-10	5.00E-07	1.14E-13	7.51E-10	1.13E-13	1.12E-08	2.31E-10
ROS	8.62E-10	3.33E-06	3.75E-12	8.23E-07	1.12E-08	1.48E-08	7.50E-10
SMOTE	7.31E-09	5.41E-05	4.21E-11	0.002	1.64E-08	1.12E-08	6.09E-11
ADASYN	3.261E-07	1.24E-05	5.52E-08	9.61E-05	3.85E-08	4.07E-08	8.26E-08
BLSMOTE	9.88E-10	3.58E-06	4.88E-12	1.65E-05	1.654E-08	1.12E-08	7.51E-10
POLYNYM-FIT-SMOTE	4.57E-09	1.54E-05	3.75E-12	0.0003	1.13E-13	1.12E-08	1.97E-10
LVQ-SMOTE	4.05E-09	6.47E-05	1.55E-11	0.0013	8.38E-08	1.13E-06	1.07E-05
G-SMOTE	2.80E-09	2.13E-05	3.49E-11	8.66E-05	1.64E-08	1.59E-08	1.91E-09
SMOBD	1.28E-08	1.03E-05	2.87E-11	0.0008	1.38E-08	1.12E-08	1.02E-10
LEE	1.03E-08	0.0001	3.65E-10	9.7E-05	1.12E-08	1.29E-08	6.52E-10
SMOTE-IPF	5.79E-09	9.02E-06	2.78E-08	0.0002	1.13E-13	1.12E-08	1.02E-10
RUS	5.15E-06	1.59E-12	6.51E-09	1.59E-12	0.81	2.09E-07	1.82E-07
TOMEK-LINK	2.31E-10	2.28E-05	7.95E-13	1.59E-07	1.13E-13	1.12E-08	2.70E-10
ENN	4.05E-09	0.00028	2.70E-10	0.0004	2.27E-13	1.15E-07	9.06E-08
NC	4.05E-09	0.0002	9.44E-08	5.41E-05	1.38E-08	3.63E-08	9.88E-10
CNN	1.47E-09	1.05E-06	1.59E-12	2.47E-06	1.02E-07	8.42E-06	7.72E-05
IHT	2.73E-07	1.97E-10	1.44E-08	6.15E-08	0.766	0.0098	0.0164
SMOTE-ENN	5.04E-08	1.64E-05	1.29E-09	0.0017	5.24E-08	0.0001	0.016
SMOTE-TOMEK	8.62E-10	9.66E-06	1.59E-12	0.0002	1.12E-08	1.12E-08	2.31E-10
ROS + RUS	3.65E-10	1.56E-06	2.80E-09	1.45E-07	1.14E-13	1.49E-08	2.18E-09
RANDOM BALANCE	8.62E-10	0.0013	9.20E-09	0.012	1.14E-13	1.65E-08	1.43E-10

Table 4.7: p-values of the Wilcoxon Signed Rank Tests for the proposed ensemble algorithm compared to other ensemble techniques

Methods	G-MEAN	MCC	ROC	F1-Score	Sensitivity	Specificity	Accuracy
OVER-BAGGING	3.49E-11	2.27E-13	2.27E-13	1.14E-13	1.39E-08	5.04E-08	1.43E-10
SMOTE-BAGGING	7.51E-10	2.27E-13	2.27E-13	1.14E-13	1.59E-08	2.46E-08	7.48E-08
BALANCED-BAGGING	1.59E-12	8.26E-08	2.08E-06	2.85E-04	7.08E-01	2.73E-02	2.11E-08
OVER-BOOSTING	2.87E-06	8.66E-11	1.48E-09	4.50E-08	7.38E-06	1.49E-03	2.61E-03
RUSBOOST	2.16E-12	1.25E-11	1.98E-10	5.45E-08	3.65E-01	9.33E-02	2.70E-10
EASY-ENSEMBLE	1.56E-11	1.79E-04	3.21E-03	7.14E-02	1.20E-07	1.43E-10	1.00E-11
BRF	1.92E-11	1.02E-02	7.11E-02	1.08E-01	1.63E-05	5.15E-09	6.09E-11

4.6 Comparative Advantages of the Proposed Method Over Alternative Approaches

A novel data resampling methodology and its ensemble counterpart have been proposed in this study. The algorithms are designed in such a way that they can overcome the data difficulty factors properly while simultaneously alleviating the limitations of the traditional sampling approaches.

To mitigate the issues caused by typical oversampling techniques, we propose a refined version of the SMOTE approach that prioritizes the quality of synthetic samples. These samples are generated near the decision boundary to better support decision-making. The samples that are far away from the decision boundary (safe zones) are ignored while generating instances. Samples that are very much inside the opposite class (noisy samples) are also overlooked and not considered for sample generation.

This ensures better representative samples and the quantity of generated samples is kept moderate to reduce overlapping.

Following this, a noise-removal technique is employed to eliminate noisy, overlapping majority-class samples from the oversampled dataset. The synthetic samples are carefully generated but this does increase overlapping with the majority-class instances. This noise-removal stage clears such overlapping regions to support classifiers in distinguishing the opposite class instances. This two-pronged approach addresses the common problems associated with oversampling. Subsequently, a random selection of majority class samples is removed to lower the IR and further decrease overlapping.

Together, these three stages achieve a balanced dataset while minimizing overlapping and noisy samples. This cohesive strategy ensures high-quality resampled data, allowing the ML model to effectively distinguish between different classes. Additionally, by balancing both undersampling and oversampling, the overall size of the data remains relatively unchanged. This proposed method is straightforward and computationally efficient, making it suitable for large datasets.

Although this hybrid sampling method yields improved results, we enhance it further by integrating it with the bagging framework to achieve better generalization. The bagging process also mitigates the effects of information loss. Overall, these combined approaches outperform other state-of-the-art sampling techniques. By adopting a unified approach, our proposed algorithm effectively addresses both class imbalance and class overlapping issues, leading to enhanced prediction performance.

An overall performance comparison of the proposed approaches with alternative approaches is illustrated in Fig. 4.6 and Fig. 4.7.

4.7 Limitations of the Proposed Approach

The proposed algorithm is designed and tested only in binary classification scenarios. However, it is equally applicable to multiclass-imbalanced scenarios. Experiments for such cases have not been conducted and therefore, not included in this manuscript.

When dealing with heavily imbalanced data with very few minority-class instances, SMOTE struggles to create meaningful samples. Consequently, in these situations, performance improvements may be minimal due to the insufficient number of representative samples in the data. Given that ML algorithms are highly data-dependent, overcoming this representation bottleneck poses a significant challenge.

To reduce overlapping, the NCL algorithm is utilized here. The approach has its limitations such as a limited number of samples being removed by the algorithm. The noisy samples that are generated by the application of SMOTE need to be more metic-

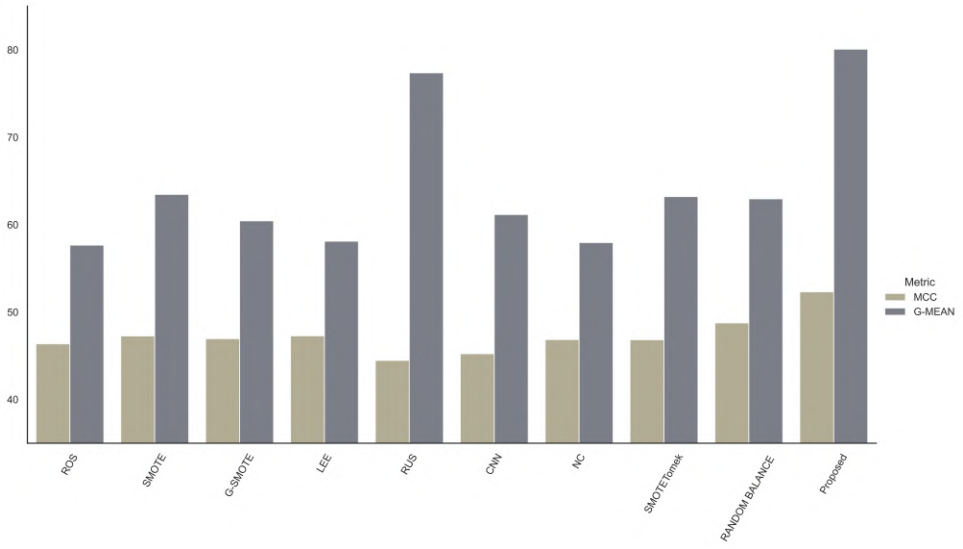

Figure 4.6: Performance comparison of the proposed approach with alternative approaches.

ulously identified and eliminated. This requires a more sophisticated methodology for removal.

4.8 Conclusion

In real-world datasets, it is common for one class to have significantly more samples than others. This imbalance causes models to favor the majority class, resulting in less accurate predictions for minority classes. Resampling the data during preprocessing is a standard method to improve performance. Numerous sampling techniques have been proposed to address this issue. However, certain intrinsic data characteristics can make the learning task challenging, and existing sampling approaches have limitations that affect their performance.

In this study, we introduce a novel sampling framework and its ensemble counterpart that can address these issues and enhance performance. Our methodology aims to reduce class overlapping, increase the presence of minority class samples in critical regions, remove noisy instances, and balance the class distribution. By integrating it with the bagging ensemble method, the model achieves better generalization.

We compared our approach with other benchmark techniques used in imbalanced learning, and our method consistently outperformed all others in terms of MCC, ROC-AUC, G-mean, and F1-score. It provides generalized performance across a wide range of imbalanced scenarios. Unlike some sampling techniques that perform well only in small to moderate imbalances, our approach excels even in highly imbalanced cases. This highlights the superiority of our proposed technique and its potential as an effec-

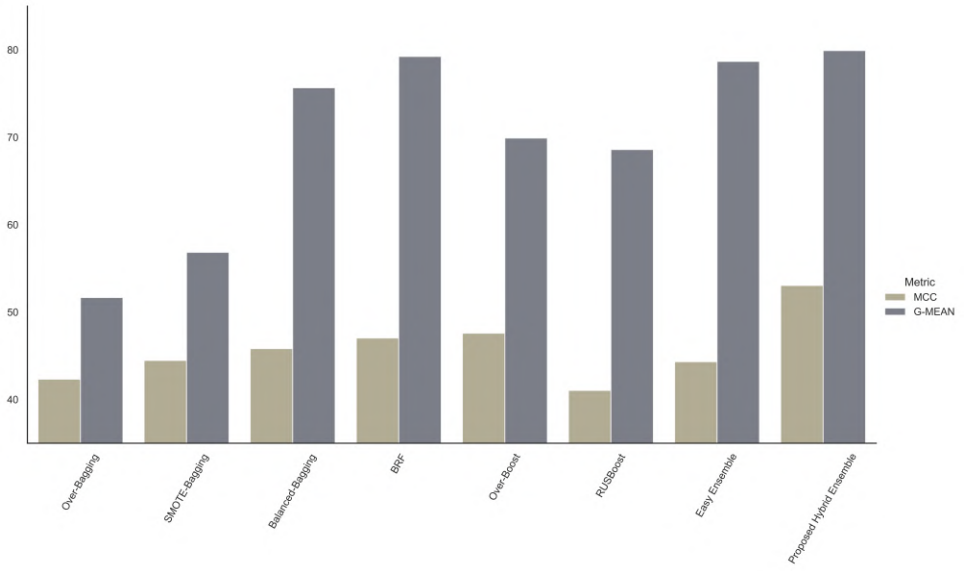

Figure 4.7: Performance comparison of the proposed ensemble approach with alternative approaches.

tive sampling method for imbalanced classification tasks.

Chapter 5

iCost: A Novel Instance Complexity Based Cost-Sensitive Learning Framework

Cost-sensitive learning is a different approach than sampling that can be used in imbalanced classification tasks. Here, the classifiers are weighted to address the class imbalance scenario. Minority-class misclassifications are highly penalized to shift the bias from the majority class. In the conventional approach, all minority-class instances are treated uniformly and assigned the same penalty value. This can have some unusual effects and overfit the data. In this thesis, a different CSL method is proposed where instances are weighted depending on their complexity level. This new methodology is described in this chapter.

5.1 Overview

In the algorithmic-level approach, the original classification algorithm is adapted to the imbalanced domain by modifying the cost function to directly address the class imbalance. This involves assigning higher misclassification costs to minority-class instances, making the algorithm more sensitive to errors involving those instances. During training, the model focuses on minimizing the overall misclassification cost. By assigning greater weight to misclassifications of the minority class, the bias is shifted away from the majority class, thus making the algorithm cost-sensitive (CS).

Not all minority-class instances present the same level of difficulty. Samples closer to the decision boundary are more likely to be misclassified than those farther away. Penalizing all the instances with the same penalty factor can deform the decision boundary creating an unusual bias towards the minority class. The model then fails to generalize well on the test data, leading to poor performance. To avoid such a scenario, more difficult-to-learn samples should be penalized more heavily. Previous literature has not considered this instance-level difficulty characteristic. Our study addresses this issue.

Our proposed algorithm addresses this issue by first categorizing all minority-class sample points based on their difficulty levels. We use a neighborhood search algorithm to grade the samples, assessing them by the number of neighboring samples from the opposite class. Higher misclassification costs are assigned to samples in overlapping regions. This prioritizes minority-class samples in these regions over majority-class samples, improving the identification of hard-to-learn instances. Conversely, samples in safe zones, surrounded by samples of the same class, are given marginal weights, reducing their impact. The appropriate penalty value can be determined through grid search, evolutionary techniques, or similar methods. This approach applies asymmetric costs to different minority-class instances based on their complexity, ensuring a more appropriate weight distribution among the minority-class samples.

5.2 Related Works

Over the years, various techniques have been proposed to address imbalanced data [13]. However, only a few of them consider data-intrinsic characteristics [87]. The idea is mostly implemented in data resampling techniques and has not been considered in CSL. However, CSL is very popular and widely applied in tackling class imbalanced scenarios [88, 89]. Different variations of the CS framework have been proposed. A comprehensive review of various CS methods is provided in this recent article [65]. A few of them are discussed below.

Gan et al. introduced a sample distribution probability-based cost-sensitive (CS) framework in their work [90]. Roychoudhuri et al. adapted the CS algorithm for time-series classification [91]. Zhou et al. extended the CS framework to handle multiclass imbalanced scenarios [92]. Other variations of CS approaches include MetaCost [93], a meta-learning algorithm that transforms any classifier into a CS classifier. The concept of example-dependent cost has also been explored in previous literature [94, 95]. For instance, in credit scoring, a borrower's credit risk is assessed based on various factors such as their credit history and financial behaviors. These factors should be considered when weighing instances for predictive modeling [96]. However, these approaches tend to be application-specific and do not generalize well to other datasets. Notably, none of these CS approaches take into account instance-difficulty-based characteristics.

5.3 Proposed Methodology

In this section, the proposed methodology is described in detail. The architecture as well as the algorithm is also depicted here.

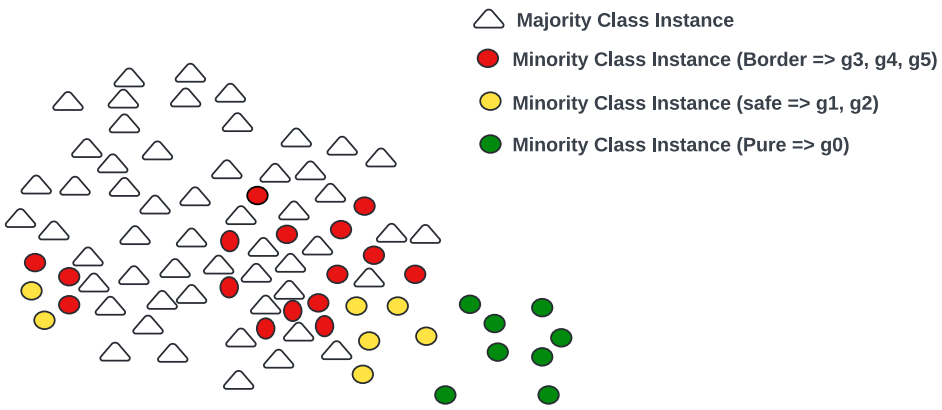

Figure 5.1: Categorization of Minority-class instances.

5.3.1 Instance Complexity

To quantify instance complexity, the K-nearest neighbors (K=5) for each minority class sample are first identified. The Euclidean distance is employed to calculate the nearest neighbors. Subsequently, each minority class instance is categorized as follows:

- **Pure**: No neighboring samples belong to the majority class.
- **Safe**: One or two neighboring samples belong to the majority class.
- **Border**: More than two neighboring samples belong to the majority class.

This categorization is depicted in Fig. 5.1. 'Pure' samples are completely surrounded by instances of their own class, making them easy to classify and typically located far from the decision boundary. Therefore, a relatively low misclassification cost is sufficient to identify these samples correctly. Assigning a higher weight could worsen the situation, leading to more misclassifications of majority class instances.

'Safe' samples have one or two neighboring instances from the opposite class and need to be handled with care due to the risk of misclassification. Too small a weight may not be adequate, while too large a weight could cause the opposite problem.

'Border' samples are surrounded by majority class instances and would be misclassified by the K-nearest neighbor classification rule. Hence, these samples require a higher weight to prioritize them over the neighboring majority-class samples.

There are alternative methods for categorizing minority-class instances. One such method, proposed by Napierala et al. [81], classifies samples into four categories: safe, borderline, rare, and outliers. This approach is incorporated into our framework by assigning four distinct penalties. We offer users greater flexibility through a general categorization formula. This formula is based on the number of majority-class samples surrounding a minority-class instance. We grade each minority-class sample from g0

to g5, corresponding to having 0 to 5 neighboring majority-class samples, respectively. Users can then assign different weights to each type of minority-class sample according to its grade.

5.3.2 Implementation

The proposed approach is implemented using the Python programming language. The theoretical framework behind CS classifiers like CS-SVM has been extensively discussed in prior literature [97], so we will not reiterate it here. The code is developed inheriting from different classifiers implementation of the sklearn library. It is fully compatible with the sklearn framework.

In our implementation, we assign distinct weights to different categories of samples according to their difficulty level. We used a grid-search technique to identify the optimal costs for each category, which varied between datasets. Based on experiments with 66 datasets, we set default values for each category to allow for immediate implementation. These default settings also notably improve performance.

Specifically, the default penalty for border samples is set to the imbalance ratio (IR) of the dataset. For safe samples, we use half of this value. For the 'pure' category, we select a cost factor of 1.2, which is almost the same as the misclassification cost for majority class samples. Using various search algorithms to find more suitable weights for individual datasets can further enhance prediction performance.

The implementation code is available in the following repository: iCost-GitHub-repository.

5.3.3 Algorithm

In this section, the framework of the algorithm is laid out in detail.

Algorithm: Instance complexity-based Cost-sensitive learning (iCost)

Inputs

- **Data**: Input dataset (Pandas DataFrame)
- **Classifier**: In this thesis, three classifier implementations have been tested. These are -
  - SVM
  - LR
  - DT

The default is set to the 'SVM' classifier. The algorithm inherits from Sklearn's SVC, LogisticRegression, and DecisionTreeClassifier implementations, respectively.

- **Type**: This refers to the type of minority-class categorization to be followed.
  - org: refers to the original implementation of the CS classifier.
  - ins: refers to the instance categorization criteria proposed in this thesis (default).
  - nap: refers to the instance categorization criteria proposed by Napierala et al. [81].
  - gen: refers to the general categorization mentioned in the previous section.
- **K**: The number of neighbors to be considered for categorization of the minority-class instances (default = 5).
- **Cost-factor:** The misclassification cost to be assigned. It can be an integer or a list/dictionary. This input parameter is related to the 'type' parameter. For type = 'org', the cost-factor value must be an integer. For other types, the cost-factor value can be both an integer and a list/dictionary. For all cases, the default value is set as the IR of the dataset.

Output

Instance-level weighted classifier fitted on the given input training data.

Procedure

- If 'type = 'org', the algorithm assigns a weight equal to the cost factor to all minority-class instances without any further considerations. This represents the original cost-sensitive implementation of the algorithms. When the cost factor value is set to 1, the algorithm functions as a standard error-driven classifier.
- When type = 'ins' or 'nap', the algorithm categorizes minority-class instances into three or four categories, respectively. For 'gen', minority-class instances are categorized into $k + 1$ categories.
- When 'type = 'ins'', the user has the option to provide either an integer or an array/dictionary with three elements as the input values for the cost factor. If an integer is provided, it is assigned as the penalty for 'border' samples. The penalty

for 'safe' samples is set to half of the integer value, and a fixed misclassification cost of 1.2 is assigned to 'pure' samples.

If an array is used as input, the values are directly assigned to 'border', 'safe', and 'pure' samples in that specific order. Alternatively, when a dictionary is provided, key-value pairs can be used to directly specify the cost values for each category.

- Similarly, for 'type = 'nap'', if the input value for the cost factor is an integer, the weights are assigned to minority class samples as follows: 'outlier' samples receive the full cost factor, 'rare' samples receive 0.75 times the cost factor, 'border' samples receive 0.5 times the cost factor, and 'safe' samples receive 0.25 times the cost factor. Alternatively, the user can directly assign weights using an array or dictionary with four elements, specifying values for 'outlier', 'rare', 'border', and 'safe' samples accordingly.

- For 'gen', the user can assign weights using an array of k+1 elements. In the case of integer input or default scenario (weight=IR), the weight is equally divided between the samples from 1 to IR proportionally based on their grade.

- The optimal values for misclassification costs vary depending on the dataset. By default, a value equivalent to the imbalance ratio (IR) of the dataset is set, akin to the approach used in the sklearn library. Assigning costs to different categories of samples, as described earlier, results in a notable improvement in performance. However, further optimization can be achieved by employing various search algorithms, as suggested in prior research [98].

- Assigning a weight lower than 1 to any minority class instance can lead to reduced sensitivity in imbalanced classification tasks, as any misclassifications of majority class samples are assigned a weight of 1. Given the greater importance of correctly classifying minority-class samples, a conditional statement is employed to ensure that the minimum weight assigned to any minority-class instance does not fall below 1. This approach helps maintain adequate sensitivity for minority class predictions in imbalanced datasets.

Example

- iCost(data, classifier = 'LR', type = 'gen', cost-factor = [5, 5, 5, 10, 10, 10])

  This will apply an instance complexity-based cost-sensitive Logistic Regression (LR) classifier on the given data. Here, g0, g1, and g2 graded minority-class samples ('pure' and 'safe' categories) are weighted by a factor of 5. The remaining samples ('border' category) are weighted by a factor of 10.

- iCost(data, classifier = 'SVM', type = 'ins', cost-factor = 20)

  This will employ the iCost algorithm with the SVM classifier. A penalty of 20 will be set for the border samples while a penalty of 10 will be set for the safe ones. The pure samples will be penalized by a factor of 1.2

5.4 Experimental Framework

5.4.1 Data

The proposed algorithm's performance was assessed across 66 imbalanced datasets, each exhibiting varying degrees of class imbalance, to validate the applicability of the approach. These datasets were sourced from the KEEL and UCI data repositories [76]. All datasets are publicly accessible and contain no missing data entries. Table 5.1 presents a summary of these datasets.

Table 5.1: Summary of the datasets

Dataset Name	# Samples	# Features	Imbalance Ratio	Dataset Name	# Samples	# Features	Imbalance Ratio
glass1	213	10	1.8	glass-0-6_vs_5	108	10	11
wisconsin	683	10	1.86	glass-0-1-4-6_vs_2	205	10	11.06
pima	768	9	1.87	glass2	214	10	11.59
glass0	213	10	2.09	ecoli-0-1-4-7_vs_5-6	332	7	12.28
yeast1	1483	9	2.46	cleveland-0_vs_4	177	14	12.62
vehicle2	846	19	2.88	shuttle-c0-vs-c4	1829	10	13.87
vehicle1	846	19	2.9	yeast-1_vs_7	459	8	14.3
vehicle3	846	19	2.99	glass4	214	10	15.46
vehicle0	845	19	3.27	ecoli4	336	8	15.8
new-thyroid1	215	6	5.14	page-blocks-1-3_vs_4	472	11	15.86
ecoli2	336	8	5.46	abalone	731	9	16.4
glass6	214	10	6.38	glass-0-1-6_vs_5	184	10	19.44
yeast3	1484	9	8.1	yeast-1-4-5-8_vs_7	693	9	22.1
yeast	1484	9	8.1	yeast4	1484	9	28.1
ecoli3	336	8	8.6	yeast128	947	9	30.57
page-blocks0	5472	11	8.79	yeast5	1484	9	32.73
ecoli-0-3-4_vs_5	200	8	9	winequality-red-8_vs_6	656	12	35.44
yeast-2_vs_4	514	9	9.08	ecoli_013vs26	281	8	39.14
ecoli-0-6-7_vs_3-5	222	8	9.09	abalone-17_vs_7-8-9-10	2338	9	39.31
ecoli-0-2-3-4_vs_5	202	8	9.1	yeast6	1483	9	41.37
yeast-0-3-5-9_vs_7-8	506	9	9.12	winequality-white-3_vs_7	900	12	44
glass-0-1-5_vs_2	172	10	9.12	winequality-red-8_vs_6-7	855	12	46.5
yeast-0-2-5-7-9_vs_3-6-8	1004	9	9.14	kddcup-land_vs_portsweep	1060	39	49.48
yeast-0-2-5-6_vs_3-7-8-9	1004	9	9.14	abalone-19_vs_10-11-12-13	1622	9	49.69
ecoli-0-4-6_vs_5	203	7	9.15	winequality_white	1481	12	58.24
ecoli-0-2-6-7_vs_3-5	224	8	9.18	poker-8-9_vs_6	1484	11	58.36
glass-0-4_vs_5	92	10	9.22	winequality-red-3_vs_5	691	12	68.1
ecoli-0-3-4-6_vs_5	205	8	9.25	abalone_20	1916	8	72.69
ecoli-0-3-4-7_vs_5-6	257	8	9.28	kddcup-land_vs_satan	1609	39	79.45
vowel	988	14	9.98	poker-8-9_vs_5	2074	11	81.96
ecoli-0-6-7_vs_5	220	7	10	poker_86	1477	11	85.88
glass-0-1-6_vs_2	192	10	10.29	kddr_rookkit	2225	42	100.14
ecoli-0-1-4-7_vs_2-3-5-6	336	8	10.59				

5.4.2 Setup

The experimental setup followed for this experiment is similar to the previous one. A repeated stratified K-fold cross-validation strategy with 5 folds and 10 repeats was adopted for this experiment to ensure more robust measurements. 3 different classifiers

(SVM, LR, and DT) were utilized. The default settings of the sklearn library were used to implement these algorithms.

For the iCost algorithm, grid-search was applied exclusively to tune the 'cost-factor' parameter. We conducted experiments using both type='org' for traditional CS classifiers and type='ins' for our modified approach. Details of the parameter configuration for the grid-search process are outlined in Table 5.2. The MCC score served as the primary criterion for evaluating performance. The choice of performance measures has been discussed in previous chapters (Chapter 2).

Table 5.2: Parameter settings for the grid-search implementation of the proposed iCost algorithm

Parameter	**Value**	**Cost-factor parameter setting**
Type	org	0.8*IR, IR, 1.2*IR
Type	ins	'pure' : [1, 0.2*IR]
		'safe': [0.25*IR, 0.35*IR, 0.5*IR]
		'border': [0.75*IR, 0.9*IR, IR, 1.1*IR, 1.25*IR]

5.4.3 Performance Comparison

To evaluate the differences, the performance of the proposed approach was compared with that of the standard CSL technique. Additionally, the results were compared with those of popular sampling techniques commonly used in imbalanced learning. The sampling techniques were implemented using the imblearn library with default parameter settings. The performance measures from these various approaches are reported in the following section.

5.5 Results and Discussion

This section contains the results obtained during the experiment. The performance of three different classifiers was measured on 66 imbalanced datasets using eight different metrics. Due to space constraints, not all these measures for individual datasets can be included here; they are provided in separate supplementary files. The average of the results on all the datasets are provided in Table 5.3, Table 5.4, and Table 5.5 for the LR, SVM, and DT classifiers, respectively.

5.5.1 Performance comparison of the proposed approach with the standard CS approach

Standard classifiers perform poorly in imbalanced data. Making them cost-sensitive improves the prediction performance significantly. This can be observed from Fig.

Table 5.3: Performance measures obtained from different approaches for the LR classifier

Metrics	LR	SMOTE	ADASYN	BL-SMOTE	ROS	RUS	ENN	NC	SMOTE_Tomek	CS-LR	iCost (Proposed)
Sensitivity	18.34	79.88	**80.81**	78.37	80.02	80.19	25.83	25.91	79.98	80.21	78.43
Specificity	**99.16**	80.46	78.16	80.77	79.70	75.16	96.40	96.30	80.44	83.26	85.47
Precision	33.64	46.70	42.75	45.56	45.46	41.65	36.05	37.31	46.69	41.87	**47.13**
ROC-AUC	58.75	80.17	79.48	79.57	79.86	77.67	61.11	61.10	80.21	81.73	**81.95**
G-mean	24.79	77.62	76.57	76.11	76.87	74.48	30.79	30.95	77.63	80.29	**81.03**
MCC	21.72	49.50	46.80	48.72	48.58	44.18	25.96	26.21	49.53	48.2	**51.56**
F1-score	21.81	52.47	49.94	51.90	51.53	47.50	27.83	28.10	52.46	50.57	**53.96**

Table 5.4: Performance measures obtained from different approaches for the SVM classifier

Metrics	SVM	SMOTE	ADASYN	BL_SMOTE	ROS	RUS	ENN	NC	SMOTE_Tomek	CS-SVM	iCost(Proposed)
Sensitivity	41.21	77.28	78.64	74.10	77.19	**83.37**	51.32	51.64	77.30	78.12	76.53
Specificity	**97.80**	86.50	84.33	87.31	85.19	76.62	94.26	94.54	86.50	89.2	91.41
Precision	53.39	55.64	52.47	56.43	54.41	44.96	53.84	55.45	55.64	55.27	**58.32**
G-mean	48.47	78.45	78.03	76.05	77.70	76.43	54.78	55.25	78.46	81.05	**82.8**
MCC	42.99	56.13	54.05	55.59	54.80	48.15	47.24	48.23	56.14	57.99	**60.02**
ROC-AUC	69.51	81.89	81.49	80.70	81.19	79.99	72.79	73.09	81.90	83.66	**83.97**
F1-Score	43.98	58.07	56.26	57.95	57.03	50.57	49.03	49.86	58.07	60.03	**61.97**

Table 5.5: Performance measures obtained from different approaches for the DT classifier

Metrics	DT	SMOTE	ADASYN	BL_SMOTE	ROS	RUS	ENN	NC	SMOTE_Tomek	CS-DT	iCost(Proposed)
Sensitivity	55.46	62.18	63.38	60.07	53.73	**81.81**	61.39	62.48	62.20	54.69	56.1
Specificity	93.61	92.02	91.69	92.59	94.50	74.48	90.44	90.62	92.00	96	**96.1**
Precision	53.60	51.95	52.15	53.05	55.00	36.37	50.31	51.54	51.78	57.38	**58.36**
G-mean	65.34	70.53	70.80	68.46	63.22	**76.29**	68.58	68.71	70.47	65.14	69.91
MCC	48.32	50.04	50.49	49.83	48.29	41.07	48.24	49.57	49.92	51.39	**52.73**
ROC-AUC	74.54	77.10	77.53	76.33	74.12	**78.14**	75.92	76.55	77.10	75.05	76.1
F1-Score	51.87	53.45	53.71	53.21	51.19	44.09	52.16	53.12	53.34	54.68	**55.9**

5.2. Fig. 5.2 illustrates the average G-mean scores obtained across 66 datasets. Of the classifiers tested, CS-SVM achieved the highest G-mean score. The DT classifier was observed to be less responsive to CS approaches. The greatest improvement in performance was observed with the LR classifier.

As compared to traditional CS approaches, in this thesis, a modified CS framework has been proposed. In this approach, different misclassification costs are assigned to minority-class instances based on their difficulty level. Samples near the decision boundary receive higher penalties compared to those that are farther away or surrounded by instances of the same class. This strategy prevents safe samples from overshadowing other majority-class instances and introducing unwarranted bias. Consequently, our proposed method offers a more plausible cost-sensitive learning framework by weighting instances according to their complexity, rather than applying a uniform approach. This modification significantly enhances performance.

The changes in the MCC score resulting from the proposed algorithm compared to the traditional CS approach are shown in Fig. 5.3, Fig. 5.4, and Fig. 5.5 for the LR, SVM, and DT classifiers, respectively. As observed in Fig. 5.3, a significant improvement is evident in most datasets for the LR classifier. A few datasets showed no change in performance, and only one dataset experienced a slight decrease in performance. For the SVM classifier, performance improved in most datasets, although the increase was smaller than that seen with the LR classifier. In several datasets, per-

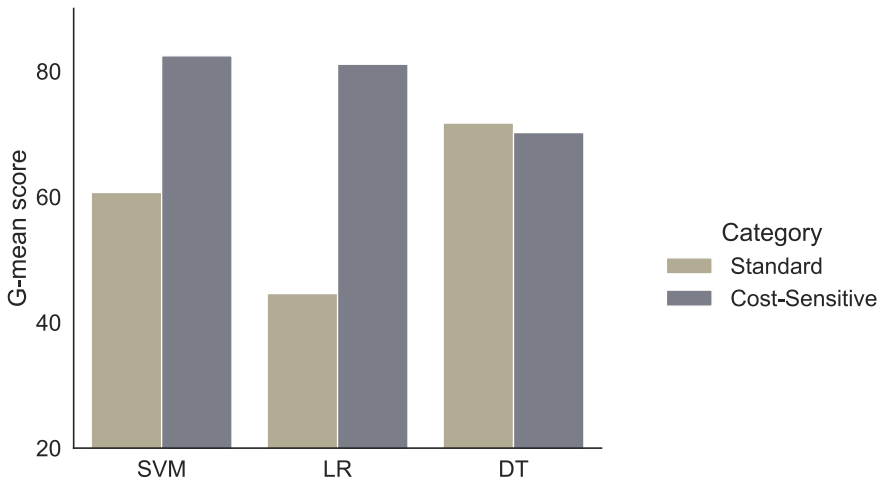

Figure 5.2: Performance comparison among standard and CS approaches.

formance remained unchanged, primarily in highly imbalanced cases (the datasets are sorted by IR in ascending order). In these cases, the limited number of minority class samples are typically surrounded by majority class instances, resulting in very few pure and safe samples. Consequently, almost all samples are border samples, which are weighted equally, causing the algorithm to behave similarly to the standard CS approach. For the DT classifier, performance declined in some datasets, though the decrease was generally minimal. In most other datasets, there was a noticeable improvement in performance.

Overall, the most significant improvement was observed with the LR classifier, averaging a 3.3% increase per dataset. For the other two classifiers, the average improvement was around 2%. MCC is a highly reliable performance metric, and an improvement in MCC indicates that our proposed method effectively reduces misclassifications. Fig. 5.6 illustrates the other performance measures (averages) obtained across 66 imbalanced datasets for the LR classifier. As shown, the proposed algorithm enhances performance across almost all measures compared to the traditional CS approach, with only a marginal drop in sensitivity. The typically small number of minority class samples in the datasets means that a few misclassifications can significantly impact the sensitivity score. However, the proposed algorithm outperformed the traditional CS approach in all four composite metrics. Similar improvements are evident for the SVM classifier as well (Table 5.4).

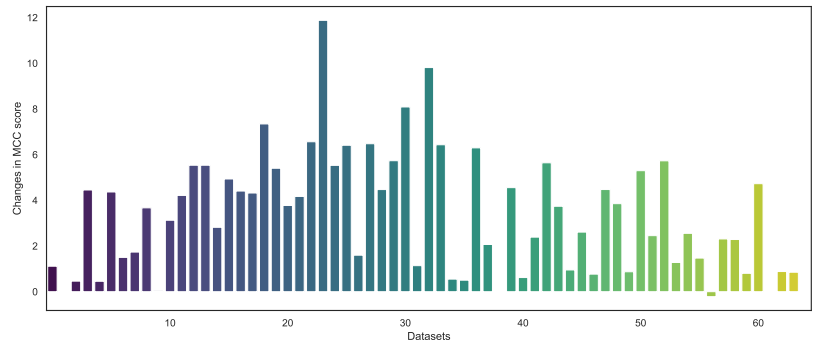

Figure 5.3: Changes in MCC score from the iCost algorithm as compared to traditional CS approach for the LR classifier on 66 datasets.

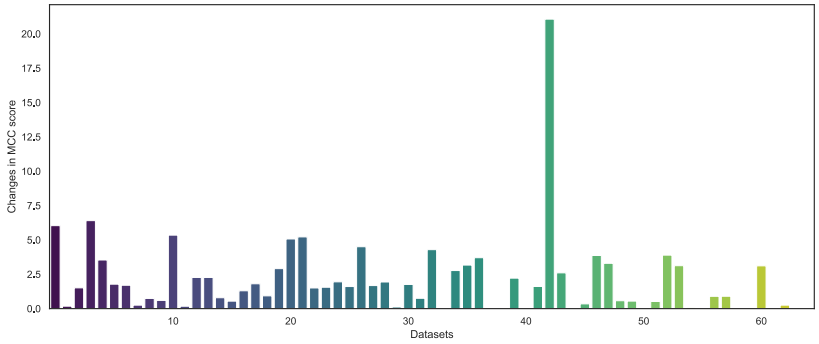

Figure 5.4: Changes in MCC score from the iCost algorithm as compared to traditional CS approach for the SVM classifier on 66 datasets.

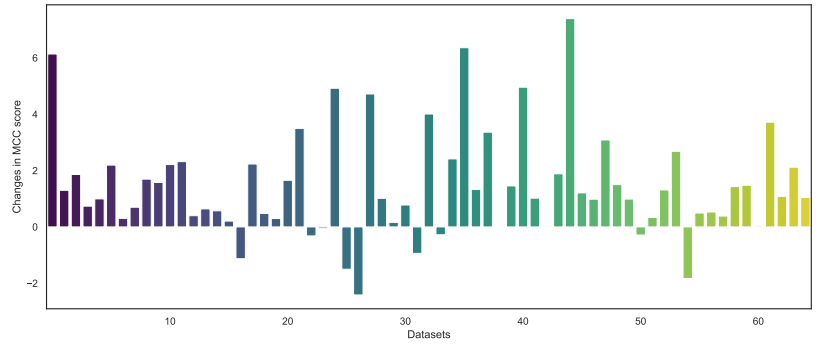

Figure 5.5: Change in MCC score from the iCost algorithm as compared to traditional CS approach for the DT classifier on 66 datasets.

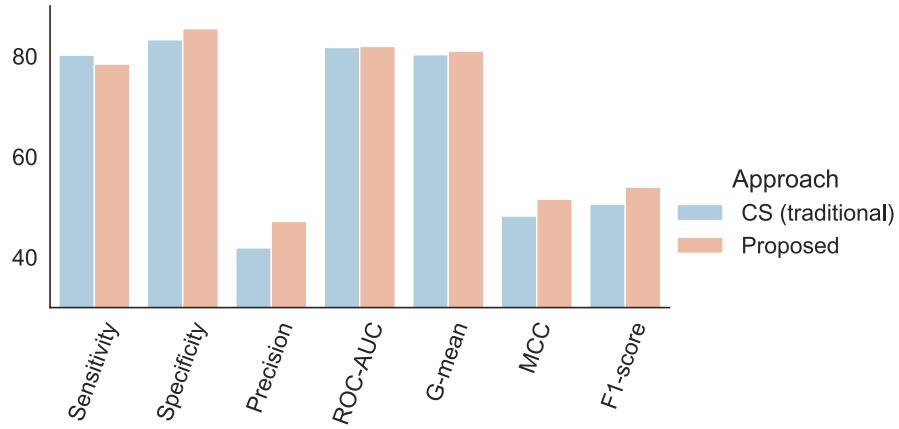

Figure 5.6: Average performance measures on 66 datasets for the LR classifier.

5.5.2 Performance comparison of the proposed algorithm with other sampling techniques

The performance was also compared with several popular sampling approaches. Data resampling techniques, which take a different approach to addressing class imbalance, have been widely used in the imbalanced domain. As shown in the tables, better prediction performance was achieved by the proposed framework compared to all other approaches in terms of precision, ROC-AUC, G-mean, MCC, and F1-score. When data was resampled using SMOTE, the most popular sampling technique, G-mean scores of 77.62% and 78.45% were obtained for the LR and SVM classifiers, respectively. This well-established method was significantly outperformed by the proposed approach, which produced G-mean scores of 81.03% and 82.8%, respectively. Significant improvements in performance were also observed compared to other approaches.

5.6 Limitations and Future Work

This study focused exclusively on binary imbalanced classification scenarios, but the concept has the potential for extension to multiclass scenarios, which we intend to explore in future research. While our study involved three different classifiers, the proposed algorithm is applicable to other classification algorithms like Random Forest or XGBoost. The default values for our approach are set empirically, necessitating further research to understand how different factors affecting data difficulty relate to the cost-factor values. Instance complexity was assessed based on nearest neighbors, although other data complexity measures such as local sets [3] could also be considered, and we plan to integrate these into our framework in future work.

5.7 Conclusion

In this study, a modified framework for Cost-sensitive learning is proposed where data difficulty factors are considered when penalizing instances. Unlike the traditional approach, where all instances are weighted equally, this method aims to address this limitation. Equal weighting can lead to biases towards the minority class, increasing false positives and potentially overfitting the data by distorting the decision boundary. This often results in higher misclassification rates during testing, especially when there is class overlap, which significantly impairs prediction performance.

To mitigate these issues, stronger weights are assigned to minority-class samples in overlapping regions compared to those in non-overlapping regions. By carefully adjusting weights to prioritize challenging examples while reducing emphasis on others, a more realistic weighting mechanism is established that minimizes misclassifications. The algorithm was evaluated on 66 imbalanced datasets using three different classifiers, demonstrating performance improvements in most cases. Importantly, the approach maintains computational efficiency similar to traditional Cost-sensitive methods. These modifications enhance the effectiveness of the Cost-sensitive framework by introducing some logical adjustments.

Chapter 6

Integrating Data Resampling and Cost-sensitive Learning: A Hybrid Approach

Data resampling and cost-sensitive learning are two popular approaches used independently for imbalanced classification tasks. These two different can be combined to achieve better prediction performance. In this chapter, such a hybrid framework combining sampling and cost-sensitive learning has been proposed.

6.1 Overview

Sampling techniques offer the advantage of independence from the underlying classification algorithm, allowing flexibility in their application with any ML classifiers. They generally improve performance compared to unsampled data, but their effectiveness can be influenced by inherent data characteristics and imbalance ratios. High IR can result in excessive generation or elimination of samples, which may lead to reduced generalization.

In contrast, cost-sensitive approaches do not alter the data distribution; they modify misclassification penalties without adding new information or reducing data complexity. However, adjusting penalties alone may not sufficiently mitigate bias from uneven class distributions. Moreover, determining the appropriate misclassification costs requires careful consideration and may vary across datasets.

While conventional practice involves using either sampling or cost-sensitive methods independently to address imbalanced scenarios, our study proposes integrating sampling techniques into the cost-sensitive learning framework. The idea is to first apply sampling to partially reduce the imbalance ratio—without fully balancing class distribution—before employing a cost-sensitive classifier. This approach avoids excessive creation or removal of minority or majority class samples, thereby reducing risks of overfitting or loss of information, respectively. By using a moderate penalty on the minority class after IR reduction, the bias from the majority class can be effectively

countered. This hybrid approach can potentially offer improved performance by mitigating complexities associated with standalone methods.

6.2 Proposed Methodology

Cost-sensitive learning alone often struggles to fully address imbalance as it does not alter the underlying imbalance in the dataset, leaving it susceptible to bias. Typically, these approaches are employed independently in practice to handle imbalanced data. However, our study introduces a novel approach that integrates both techniques to potentially enhance performance. The rationale behind this integration is to leverage the strengths of each approach while mitigating their respective drawbacks. This is achieved by first reducing the imbalance ratio through sampling, followed by applying a cost-sensitive classifier with moderate penalties assigned to the minority class.

One challenge with generating synthetic minority-class samples is that they may not accurately represent the true minority-class distribution. Similarly, removing too many majority-class samples risks losing important information. As the imbalance ratio decreases, the need for extensive sample generation or removal diminishes, reducing the risk of overfitting. By maintaining a moderate imbalance ratio and applying appropriate cost penalties, the bias towards the majority class can be effectively countered.

Achieving an optimal balance between these approaches requires tuning two key parameters: the sampling ratio (α) and the weight factor (ω). These parameters dictate how sampling and cost penalties are applied to achieve the desired balance. Below is a detailed outline of the step-by-step process for developing such a hybrid model.

Outline of the Algorithm

The processed dataset is first divided into training and testing sets to prevent data leakage, with only the training set undergoing resampling. SMOTE was chosen as the sampling approach. Other variants of SMOTE can also be utilized.

Instead of completely balancing the dataset, a controlled degree of imbalance was maintained during resampling, governed by the α parameter available in the imblearn library's SMOTE implementation. Determining the optimal α value required a search approach tailored to the respective dataset.

Following this, an XGBoost classifier was trained on the resampled dataset, with adjustments made to the 'scale_pos_weight' parameter to ensure the classifier was cost-sensitive. Both the sampling ratio (α) and the weight assigned to the minority class (ω) were concurrently tuned using grid-search. This process involved evaluating numerous potential values, which could be computationally intensive. To streamline

this, insights from previous experiments were utilized to strategically narrow down the search space and reduce computational overhead.

From these experiments, it was observed that the most effective value for the class weight parameter ω typically aligned closely with the IR of the dataset. Similarly, α values below 0.6 were insufficient in addressing bias, necessitating consideration of values above 0.6 to achieve adequate imbalance mitigation. Based on these insights, parameters for the grid search were selected to maximize the MCC score.

The parameter settings for the grid search approach are reported in Table 6.1.

Table 6.1: Parameter settings for the grid-search implementation of the proposed hybrid algorithm

Parameter	**parameter setting**
α	0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 1
ω	0.25*IR, 0.5*IR, 0.6*IR, 0.75*IR, 0.9*IR, IR

6.3 Experimental Framework

In this study, the performance of our proposed approach was assessed across an additional 36 imbalanced datasets sourced from the KEEL repository [76]. Details of these datasets are summarized in Table 6.2. Performance was compared with those of other well-known techniques in imbalanced learning. The XGBoost classifier was selected as the base learning algorithm. Other classifiers can also be utilized. A 10-fold stratified cross-validation methodology was adopted to evaluate model performance. Prior to training, the data underwent normalization to ensure consistent scaling across features. The outline of the experimental framework is demonstrated in Fig. 6.1.

6.4 Results and Discussion

The detailed performance metrics for each dataset can be found in the Additional file. Table 6.3 presents the average performance scores across the 36 imbalanced datasets. Figure 6.2 provides a comparative analysis of different approaches.

From Figure 3, it is evident that our proposed approach achieves the highest average scores in terms of ROC-AUC, g-mean, and MCC among all the techniques evaluated. The hybrid approach combining SMOTE with a weighted classifier significantly outperforms using SMOTE or a weighted classifier independently. It also outperformed other sampling techniques. There was minimal difference observed between the performance of SMOTE and ADASYN. In contrast, the RUS algorithm yielded the highest sensitivity score but at the expense of the lowest specificity and MCC scores. This imbalance in prediction performance is reflected in the composite metrics such as

Table 6.2: Summary of the datasets

Name	# samples	# positive samples	# features	IR
Wisconsin	683	239	9	1.86
vehicle2	846	218	18	2.88
vehicle1	846	217	18	2.9
vehicle3	846	212	18	2.99
vehicle0	846	199	18	3.25
new_thyroid1	215	35	5	5.14
ecoli2	336	52	7	5.46
glass6	214	29	9	6.38
yeast3	1484	163	10	8.1
ecoli3	336	35	7	8.6
yeast-2_vs_4	514	51	8	9.08
yeast-0-2-5-6_vs_3-7-8-9	1004	99	10	9.14
vowel	988	90	13	9.98
led7digit-0-2-4-6-7-89_vs_1	443	37	7	10.97
glass2	214	17	9	11.59
ecoli-0-1-4-7_vs_5-6	332	25	6	12.28
glass4	214	13	9	15.46
ecoli4	336	20	7	15.8
page-blocks-1-3_vs_4	472	28	10	15.86
abalone	731	42	8	16.4
yeast-1-4-5-8_vs_7	693	30	10	22.1
yeast	1484	51	10	28.1
yeast-1-2-8-9_vs_7	947	30	10	30.57
yeast5	1484	44	10	32.73
winequality-red-8_vs_6	656	18	11	35.44
abalone_17_vs_7_8_9_10	2338	58	8	39.31
winequality-white-3_vs_7	900	20	11	44
winequality-red-8_vs_6-7	855	18	11	46.5
Kddcup land_vs_portsweep	1061	21	40	49.52
abalone-19_vs_10-11-12-13	1622	32	8	49.69
winequality-white-3-9_vs_5	1482	25	11	58.28
poker-8-9_vs_6	1485	25	25	58.4
winequality-red-3_vs_5	691	10	11	68.1
kddcup-land_vs_satan	1610	21	30	75.67
poker-8-9_vs_5	2075	25	25	82
poker-8_vs_6	1477	17	25	85.88

MCC. Our proposed approach consistently achieved superior scores across these metrics. It provided the highest MCC, G-mean, ROC-AUC as well as precision score. The performance measures demonstrate its effectiveness in addressing class imbalance.

Table 6.3: Average of the performance measures obtained from different approaches on 36 imbalanced datasets

Metrics	SMOTE	ADASYN	RUS	Tomek-link	ENN	Weighted XGBoost	Proposed
Accuracy	93.57	94.49	78.09	94.25	93.64	94	93.3
Sensitivity	62.34	62.5	79.72	51.06	56.79	56.44	71.22
Specificity	95.22	95.09	77.87	96.46	94.92	95.91	94.35
G-mean	70.3	70.54	76.45	59.22	62.76	65.18	78.52
ROC-AUC	78.78	78.79	78.8	73.76	75.86	76.17	82.78
Precision	59.11	59.93	39.64	57.64	55.07	59.61	60.09
MCC	56.14	56.57	46.62	49.94	51.24	53.49	60.55

6.5 Limitations and Future Work

This approach was conducted initially with the standard sampling and CS approaches to understand the viability of such hybridization. Good improvement in performance was noticeable. Other more sophisticated sampling and CS techniques are equally applicable to form such hybridization. These can further improve the prediction performance.

The algorithm has been tested only in binary classification scenarios. It can also be extended for multiclass classification. Only the XGBoost classifier was utilized as the base learner. Other classification algorithms can also be utilized in its place. Only the SMOTE algorithm was tested for sampling the data in the proposed approach. Other sampling techniques as well as the proposed unified sampling framework presented in this thesis (Chapter 4) can also be utilized. Using undersampling techniques to lower the IR while increasing the weights of the minority-class instances can be a viable option, which we plan to test in future works. The modified CS framework, iCost (Chapter 5), can also be utilized for the weighted classifier.

6.6 Conclusion

In this study, a novel approach to addressing class imbalance is proposed, combining sampling techniques with a cost-sensitive learning framework. The hypothesis is that an optimal balance between these two methods can enhance prediction performance. This hybrid technique requires generating fewer minority-class samples and discarding fewer majority-class samples to achieve balance. Furthermore, it reduces the penalty weight needed in the cost-sensitive classifier.

This method effectively mitigates the limitations associated with both sampling techniques and cost-sensitive approaches. The SMOTE algorithm was employed for sampling in this study due to its superior performance, though other sampling techniques can be integrated into the proposed framework. The results demonstrate that the proposed approach outperformed traditional sampling techniques and cost-sensitive

learning in both ROC-AUC and MCC scores, indicating its effectiveness in addressing imbalanced classification problems.

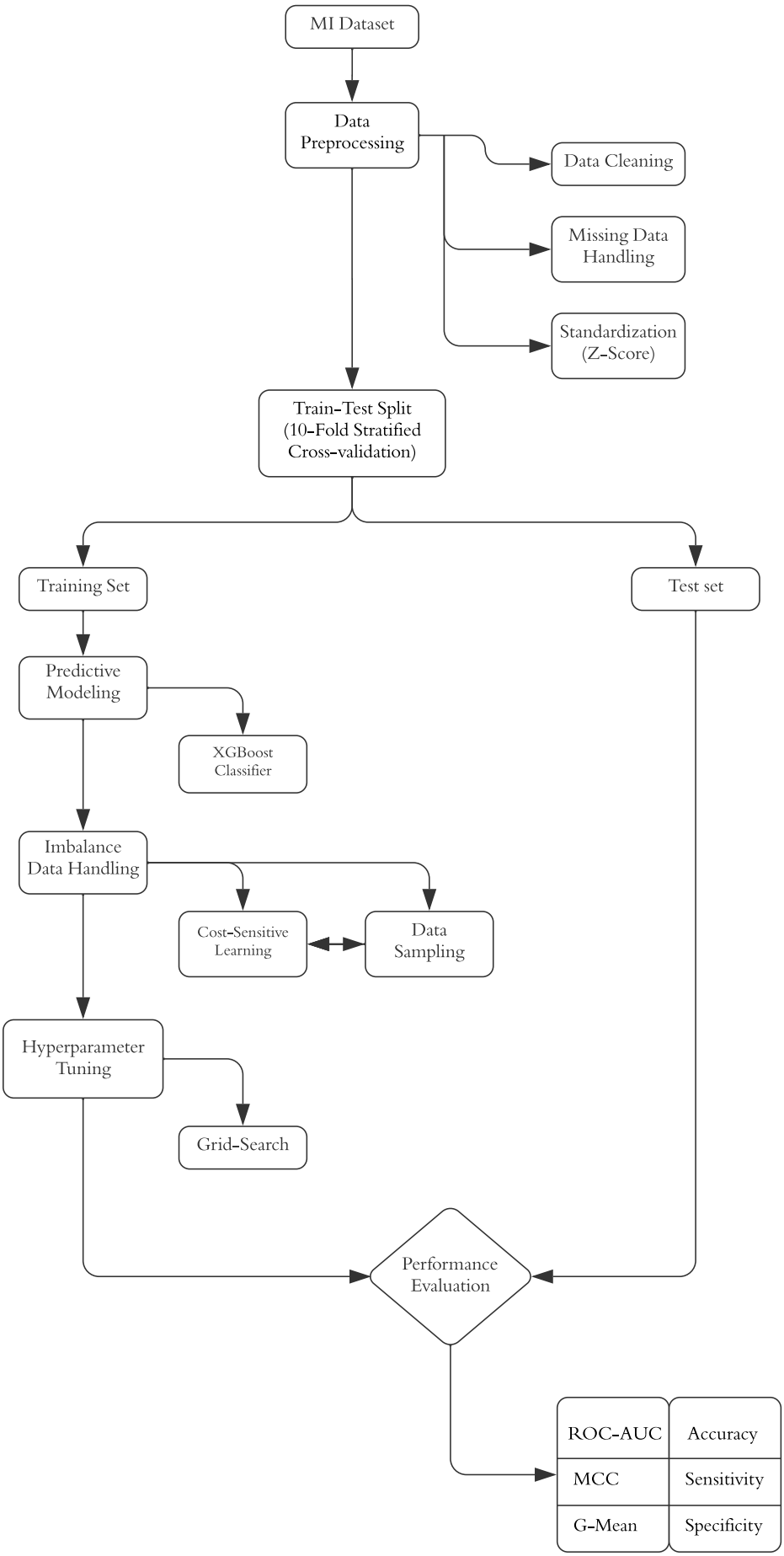

Figure 6.1: Outline of the experimental framework.

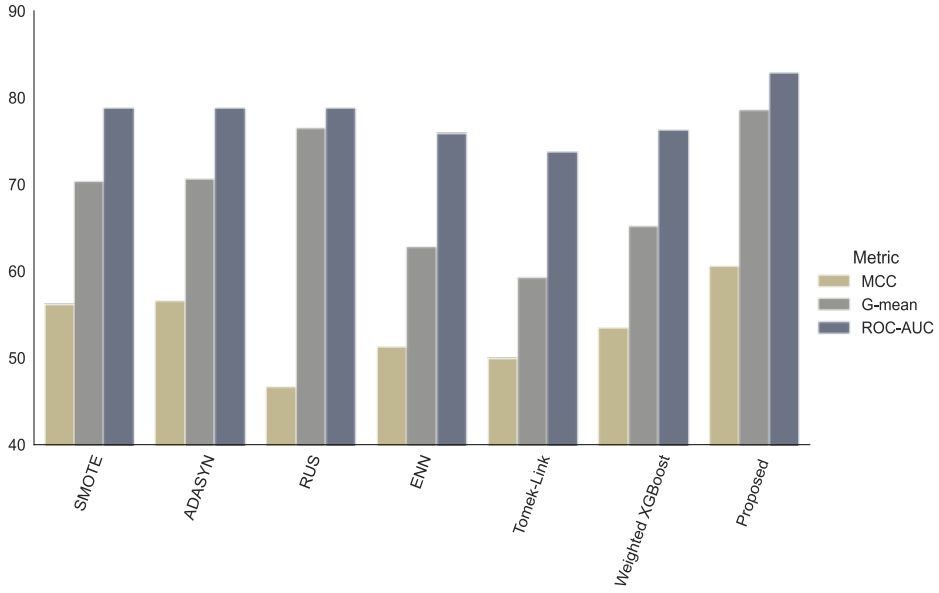

Figure 6.2: Performance comparison with other approaches.

Chapter 7

Conclusion

The works conducted as a part of this thesis are summarized in this chapter. The ongoing research work on this topic as well as future works is also discussed here.

7.1 Ongoing Research Work

Most of the studies focus on binary imbalanced classification scenarios. However, data with multiple classes are also quite prevalent but the imbalanced data handling techniques proposed in research often neglect these cases. Multiclass imbalanced classification scenarios are far more complicated than binary ones and require more careful handling. This has been discussed in these recent articles [68, 99].

The most commonly used approach to tackle multiclass classification is the decomposition strategy [100]. Here, the data with multiple classes is divided into binary classes through certain steps. Then, the traditional approaches for binary classification can be utilized. Finally, the predictions are combined to obtain the overall performance. One-vs-One (OVO) and One-vs-All (OVA) are two such different decomposition strategies [101]. Both are widely used in tackling different real-world classification problems.

This type of decomposition creates certain issues. One of the issues with such a decomposition strategy is that the resultant data with binary classes can become quite imbalanced even if the original classes are even. The problem becomes far more complicated when the data is imbalanced, which is more common in different applications. To give an example, let us consider a dataset with 3 classes. The number of instances available on this data are 100, 2000, and 10000 for three different classes. Now, applying the OVA strategy will create a scenario where the classifier has to learn from a dataset with 100 instances from one class vs 12000 instances from the other class. The class imbalance that is already present is enhanced by the application of such a

decomposition strategy. The situation gets even more complex with a higher number of categories. Proper steps need to be taken to alleviate such a scenario.

Several different strategies have been proposed by researchers to tackle multiclass classification scenarios [102–104]. The traditional sampling or CS approaches are equally applicable to multiclass scenarios. However, they do not address the other complications that arise when the data has multiple categories. The decomposition strategies divide the data without considering any other factors such as class imbalance or overlapping. This leads to very poor performance in minority class categories.

To alleviate the scenario, we propose a novel decomposition strategy where the classification is performed in multiple stages and the data is strategically divided, taking into consideration both the class imbalance and overlapping. Here, in the earlier stages, classification is performed for the large classes while in subsequent stages, classification is performed for smaller classes. This way, the issues faced during decomposition are alleviated and improved prediction performance can be achieved.

The proposed method is currently under development with additional strategies being added for improved performance and generalization. The initial results are promising and the method is found to be quite effective in handling multiclass imbalanced scenarios.

7.2 Future Work

As a continuation of the work presented in this thesis, we plan to further explore diverse issues in the imbalanced domain. They are as follows.

- Deep learning (DL) techniques continue to evolve rapidly, with increased application on a wide range of problems [105–109]. These techniques are also applicable in imbalanced learning and have been applied in some recent works [110, 111]. More specifically, Generative Adversarial Networks (GAN) have been utilized to generate synthetic samples. Deep Reinforcement Learning (DRL) has also been utilized for imbalanced classification tasks [64]. A major issue with the application of DL techniques in tabular data is that they require large datasets which is usually unavailable in many cases, especially in critical applications such as healthcare [112, 113]. We plan to explore the feasibility of such techniques in diverse imbalanced cases in the future.

- We plan to explore the effect of different sampling techniques on class overlapping on a broad range of real-world imbalanced datasets. Relating their effect on

class overlapping with the performance of those approaches will provide greater insight and help develop more robust approaches.

- We want to extend and modify the proposed iCost algorithm to deal with multi-class imbalanced scenarios in the future. We hypothesize that incorporating class overlapping issues into the selection of penalty factors can improve performance in rare classes.

- Creation of a new repository for imbalanced datasets is imperative. Existing ones are quite outdated with no further inclusion of new datasets. A new repository featuring a diverse array of real-world datasets would greatly benefit researchers worldwide.

7.3 Summary

This thesis deals with the imbalanced learning problem that is frequently faced in different real-world classification tasks. As the standard classification algorithms become biased when the data is skewed, necessary steps need to be taken to obtain reliable predictions. The problem has caught the attention of researchers and different strategies have been developed over the years to address this issue. Recent studies call for new directions in this domain to overcome some of the bottlenecks and improve the performance of traditional methods. In that regard, this thesis dives into new frontiers in the imbalanced domain, especially focusing on data characteristics. Different data complexity issues have been analyzed and novel strategies have been developed to overcome the limitations of the existing methods and enhance prediction performance.

First of all, an extensive experimental analysis of the popular state-of-the-art methods used in imbalanced learning has been conducted on a wide variety of datasets. The strengths and weaknesses of the algorithms have been identified. The major issues that affect the performance of the classifiers when the data is imbalanced have been distinguished. A thorough literature review has also been conducted to understand the current status. It has been observed that the current strategies lack certain adaptability. Many of those techniques work well only when the imbalance is low. They fail tragically in higher imbalances. The class overlapping issue, which has been identified as one of the major causes of data complexity, has been considered by a handful of approaches only. Other data difficulty factors such as the rarity of certain classes, the presence of noisy samples, and small disjunct have not been properly addressed by most strategies. While some techniques take care of one issue but fail to address the others. Consequently, they lack performance and generalization over a wide range of

data. Based on these analyses and observations, three new methods have been proposed in this study.

The first method is a unified approach designed to address both class imbalance and class overlapping simultaneously. A modified SMOTE algorithm is utilized to ensure the proper generation of minority-class samples. A noise removal filter and an undersampling technique are also added to further address other data difficulty factors. The proposed sampling technique is then integrated into a modified bagging ensemble framework for better generalization. The proposed approaches have been tested on different imbalanced scenarios and a significant improvement in performance over other state-of-the-art sampling techniques has been observed. The algorithm also performs quite well in higher imbalances where traditional approaches fail.

Secondly, a modified cost-sensitive approach is proposed that considers data complexity factors. In the traditional approach, all instances are treated equally and penalized. In the proposed method, instances are categorized based on their difficulty level and penalized accordingly. This ensures a more plausible weighting of instances and produces better prediction performance.

Thirdly, a hybridization of data resampling and cost-sensitive learning approaches has been proposed. Such a hybrid method reduces the need for excessive sampling and penalization. This type of approach also showed improved prediction performance and can be useful while handling imbalanced data.

In addition to these, ongoing research is focused on multiclass imbalanced scenarios. A novel decomposition strategy for multiclass datasets is currently being developed.

To conclude, this thesis presents novel approaches that consider data complexity factors to address the imbalanced scenario. The proposed approaches have been proven to be quite successful and outperform the state-of-the-art techniques in different imbalanced scenarios. This gives a new direction in the imbalanced domain and paves the way for the development of new strategies that are based on data-specific characteristics.

REFERENCES

[1] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM computing surveys (CSUR)*, vol. 49, no. 2, pp. 1–50, 2016.

[2] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018, vol. 10, no. 2018.

[3] M. S. Santos, P. H. Abreu, N. Japkowicz, A. Fernández, C. Soares, S. Wilk, and J. Santos, "On the joint-effect of class imbalance and overlap: a critical review," *Artificial Intelligence Review*, vol. 55, no. 8, pp. 6207–6275, 2022.

[4] M. Mercier, M. S. Santos, P. H. Abreu, C. Soares, J. P. Soares, and J. Santos, "Analysing the footprint of classifiers in overlapped and imbalanced contexts," in *International symposium on intelligent data analysis*. Springer, 2018, pp. 200–212.

[5] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[6] F. Shen, Y. Liu, R. Wang, and W. Zhou, "A dynamic financial distress forecast model with multiple forecast results under unbalanced data environment," *Knowledge-Based Systems*, vol. 192, p. 105365, 2020.

[7] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, p. 107377, 2020.

[8] A. Newaz, N. Ahmed, and F. S. Haq, "Survival prediction of heart failure patients using machine learning techniques," *Informatics in Medicine Unlocked*, vol. 26, p. 100772, 2021.

[9] M. Dudjak and G. Martinović, "An empirical study of data intrinsic characteristics that make learning from imbalanced data difficult," *Expert systems with applications*, vol. 182, p. 115297, 2021.

[10] M. S. Mohosheu, M. A. al Noman, A. Newaz, T. Jabid *et al.*, "A comprehensive evaluation of sampling techniques in addressing class imbalance across diverse datasets," in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*. IEEE, 2024, pp. 1008–1013.

[11] A. Newaz, M. S. Mohosheu, M. A. Al Noman, and T. Jabid, "ibrf: Improved balanced random forest classifier," in *2024 35th Conference of Open Innovations Association (FRUCT)*. IEEE, 2024, pp. 501–508.

[12] A. Newaz, M. S. Mohosheu, and M. A. Al Noman, "Predicting complications of myocardial infarction within several hours of hospitalization using data mining techniques," *Informatics in Medicine Unlocked*, vol. 42, p. 101361, 2023.

[13] S. Rezvani and X. Wang, "A broad review on class imbalance learning techniques," *Applied Soft Computing*, p. 110415, 2023.

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

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

[16] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. Ieee, 2008, pp. 1322–1328.

[17] G. Kovács, "smote-variants: a python implementation of 85 minority oversampling techniques," *Neurocomputing*, vol. 366, pp. 352–354, 2019, (IF-2019=4.07).

[18] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[19] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13*. Springer, 2009, pp. 475–482.

[20] ——, "Dbsmote: density-based synthetic minority over-sampling technique," *Applied Intelligence*, vol. 36, pp. 664–684, 2012.

[21] L. Ma and S. Fan, "Cure-smote algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests," *BMC bioinformatics*, vol. 18, pp. 1–18, 2017.

[22] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 2, pp. 405–425, 2012.

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

[24] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," *Data mining and knowledge discovery*, vol. 28, pp. 92–122, 2014.

[25] G. Douzas and F. Bacao, "Geometric smote a geometrically enhanced drop-in replacement for smote," *Information sciences*, vol. 501, pp. 118–135, 2019.

[26] B. A. Almogahed and I. A. Kakadiaris, "Neater: filtering of over-sampled data using non-cooperative game theory," *Soft Computing*, vol. 19, pp. 3301–3322, 2015.

[27] M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognition*, vol. 45, no. 10, pp. 3738–3750, 2012.

[28] H.-J. Kim, N.-O. Jo, and K.-S. Shin, "Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction," *Expert systems with applications*, vol. 59, pp. 226–234, 2016.

[29] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 3, pp. 408–421, 1972.

[30] P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE transactions on information theory*, vol. 14, no. 3, pp. 515–516, 1968.

[31] H. Yu, J. Ni, and J. Zhao, "Acosampling: An ant colony optimization-based undersampling method for classifying imbalanced dna microarray data," *Neurocomputing*, vol. 101, pp. 309–318, 2013.

[32] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," *Machine learning*, vol. 95, pp. 225–256, 2014.

[33] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[34] J. F. Díez-Pastor, J. J. Rodriguez, C. Garcia-Osorio, and L. I. Kuncheva, "Random balance: ensembles of variable priors classifiers for imbalanced data," *Knowledge-Based Systems*, vol. 85, pp. 96–111, 2015.

[35] J. J. Rodriguez, J.-F. Diez-Pastor, A. Arnaiz-Gonzalez, and L. I. Kuncheva, "Random balance ensembles for multiclass imbalance learning," *Knowledge-Based Systems*, vol. 193, p. 105434, 2020.

[36] A. Newaz, S. Muhtadi, and F. S. Haq, "An intelligent decision support system for the accurate diagnosis of cervical cancer," *Knowledge-Based Systems*, vol. 245, p. 108634, 2022.

[37] C. Chen, A. Liaw, L. Breiman *et al.*, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, no. 1-12, p. 24, 2004.

[38] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.

[39] L. Nanni, C. Fantozzi, and N. Lazzarini, "Coupling different methods for overcoming the class imbalance problem," *Neurocomputing*, vol. 158, pp. 48–61, 2015.

[40] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern recognition*, vol. 46, no. 12, pp. 3460–3471, 2013.

[41] Q. Cao and S. Wang, "Applying over-sampling technique based on data density and cost-sensitive svm to imbalanced learning," in *2011 International Conference on Information Management, Innovation Management and Industrial Engineering*, vol. 2. IEEE, 2011, pp. 543–548.

[42] S. S. Mullick, S. Datta, S. G. Dhekane, and S. Das, "Appropriateness of performance indices for imbalanced data classification: An analysis," *Pattern Recognition*, vol. 102, p. 107197, 2020.

[43] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data–recommendations for the use of performance metrics," in *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE, 2013, pp. 245–251.

[44] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, pp. 1–13, 2020.

[45] Q. Zhu, "On the performance of matthews correlation coefficient (mcc) for imbalanced dataset," *Pattern Recognition Letters*, vol. 136, pp. 71–80, 2020.

[46] P. Vuttipittayamongkol and E. Elyan, "Overlap-based undersampling method for classification of imbalanced medical datasets," in *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 358–369.

[47] A. Newaz, S. Hassan, and F. S. Haq, "An empirical analysis of the efficacy of different sampling techniques for imbalanced classification," *arXiv preprint arXiv:2208.11852*, 2022.

[48] P. Vuttipittayamongkol, E. Elyan, and A. Petrovski, "On the class overlap problem in imbalanced data classification," *Knowledge-based systems*, vol. 212, p. 106631, 2021.

[49] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Artificial Intelligence in Medicine: 8th Conference on Artificial Intelligence in Medicine in Europe, AIME 2001 Cascais, Portugal, July 1–4, 2001, Proceedings 8*. Springer, 2001, pp. 63–66.

[50] C. Bunkhumpornpat and K. Sinapiromsaran, "Dbmute: density-based majority under-sampling technique," *Knowledge and Information Systems*, vol. 50, pp. 827–850, 2017.

[51] P. Vuttipittayamongkol, E. Elyan, A. Petrovski, and C. Jayne, "Overlap-based undersampling for improving imbalanced data classification," in *Intelligent Data Engineering and Automated Learning–IDEAL 2018: 19th International Conference, Madrid, Spain, November 21–23, 2018, Proceedings, Part I 19*. Springer, 2018, pp. 689–697.

[52] D. Devi, B. Purkayastha *et al.*, "Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance," *Pattern Recognition Letters*, vol. 93, pp. 3–12, 2017.

[53] E. R. Fernandes and A. C. de Carvalho, "Evolutionary inversion of class distribution in overlapping areas for multi-class imbalanced learning," *Information Sciences*, vol. 494, pp. 141–154, 2019.

[54] J. Lee, N.-r. Kim, and J.-H. Lee, "An over-sampling technique with rejection for imbalanced class learning," in *Proceedings of the 9th international conference on ubiquitous information management and communication*, 2015, pp. 1–6.

[55] S. Gazzah and N. E. B. Amara, "New oversampling approaches based on polynomial fitting for imbalanced data sets," in *2008 the eighth iapr international workshop on document analysis systems*. IEEE, 2008, pp. 677–684.

[56] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evolutionary computation*, vol. 17, no. 3, pp. 275–306, 2009.

[57] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409, pp. 17–26, 2017.

[58] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in *2009 IEEE symposium on computational intelligence and data mining*. IEEE, 2009, pp. 324–331.

[59] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7*. Springer, 2003, pp. 107–119.

[60] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*, vol. 40, no. 1, pp. 185–197, 2009.

[61] M. S. Santos, P. H. Abreu, N. Japkowicz, A. Fernández, and J. Santos, "A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research," *Information Fusion*, vol. 89, pp. 228–253, 2023.

[62] A. S. Tarawneh, A. B. Hassanat, G. A. Altarawneh, and A. Almuhaimeed, "Stop oversampling for class imbalance learning: A review," *IEEE Access*, vol. 10, pp. 47 643–47 660, 2022.

[63] A. Guzmán-Ponce, R. M. Valdovinos, J. S. Sánchez, and J. R. Marcial-Romero, "A new under-sampling method to face class overlap and imbalance," *Applied Sciences*, vol. 10, no. 15, p. 5164, 2020.

[64] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Applied Intelligence*, vol. 50, no. 8, pp. 2488–2502, 2020.

[65] G. Petrides and W. Verbeke, "Cost-sensitive ensemble learning: a unifying framework," *Data Mining and Knowledge Discovery*, vol. 36, no. 1, pp. 1–28, 2022.

[66] J. Stefanowski, "Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data," in *Emerging paradigms in machine learning*. Springer, 2013, pp. 277–306.

[67] V. H. Barella, L. P. Garcia, M. C. de Souto, A. C. Lorena, and A. C. de Carvalho, "Assessing the data complexity of imbalanced datasets," *Information Sciences*, vol. 553, pp. 83–109, 2021.

[68] M. Lango and J. Stefanowski, "What makes multi-class imbalanced problems difficult? an experimental study," *Expert Systems with Applications*, vol. 199, p. 116962, 2022.

[69] M. Denil and T. Trappenberg, "Overlap versus imbalance," in *Advances in Artificial Intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010, Ottawa, Canada, May 31–June 2, 2010. Proceedings 23*. Springer, 2010, pp. 220–231.

[70] R. C. Prati, G. E. Batista, and M. C. Monard, "Class imbalances versus class overlapping: an analysis of a learning system behavior," in *MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, April 26-30, 2004. Proceedings 3*. Springer, 2004, pp. 312–321.

[71] Z. Borsos, C. Lemnaru, and R. Potolea, "Dealing with overlap and imbalance: a new metric and approach," *Pattern Analysis and Applications*, vol. 21, pp. 381–395, 2018.

[72] R. C. Prati, G. E. Batista, and M. C. Monard, "Learning with class skews and small disjuncts," in *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17*. Springer, 2004, pp. 296–306.

[73] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 40–49, 2004.

[74] G. M. Weiss, "The impact of small disjuncts on classifier learning," in *Data Mining: Special Issue in Annals of Information Systems*. Springer, 2009, pp. 193–226.

[75] M. Kelly, R. Longjohn, and K. Nottingham, "The uci machine learning repository (2023)," *URL https://archive. ics. uci. edu*, 2023.

[76] J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Mult. Valued Logic Soft Comput*, vol. 17, pp. 255–287, 2015.

[77] S. Oh, "A new dataset evaluation method based on category overlap," *Computers in Biology and Medicine*, vol. 41, no. 2, pp. 115–122, 2011.

[78] B. Omar, F. Rustam, A. Mehmood, G. S. Choi *et al.*, "Minimizing the overlapping degree to improve class-imbalanced learning under sparse feature selection: application to fraud detection," *IEEE Access*, vol. 9, pp. 28 101–28 110, 2021.

[79] S. Mayabadi and H. Saadatfar, "Two density-based sampling approaches for imbalanced and overlapping data," *Knowledge-Based Systems*, vol. 241, p. 108217, 2022.

[80] D. Devi, S. K. Biswas, and B. Purkayastha, "Learning in presence of class imbalance and class overlapping by using one-class svm and undersampling technique," *Connection Science*, vol. 31, no. 2, pp. 105–142, 2019.

[81] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," *Journal of Intelligent Information Systems*, vol. 46, pp. 563–597, 2016.

[82] F. Wilcoxon, S. Katti, R. A. Wilcox *et al.*, "Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test," *Selected tables in mathematical statistics*, vol. 1, pp. 171–259, 1970.

[83] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Applied Soft Computing*, vol. 83, p. 105662, 2019.

[84] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[85] M. Nakamura, Y. Kajiwara, A. Otsuka, and H. Kimura, "Lvq-smote–learning vector quantization based synthetic minority over–sampling technique for biomedical data," *BioData mining*, vol. 6, pp. 1–10, 2013.

[86] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 2, no. 5-6, pp. 412–426, 2009.

[87] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert systems with applications*, vol. 73, pp. 220–239, 2017.

[88] A. Newaz, N. Ahmed, and F. S. Haq, "Diagnosis of liver disease using cost-sensitive support vector machine classifier," in *2021 International Conference on Computational Performance Evaluation (ComPE)*. IEEE, 2021, pp. 421–425.

[89] Y. Yang, S. Huang, W. Huang, and X. Chang, "Privacy-preserving cost-sensitive learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2105–2116, 2020.

[90] D. Gan, J. Shen, B. An, M. Xu, and N. Liu, "Integrating tanbn with cost sensitive classification algorithm for imbalanced data in medical diagnosis," *Computers & Industrial Engineering*, vol. 140, p. 106266, 2020.

[91] S. Roychoudhury, M. Ghalwash, and Z. Obradovic, "Cost sensitive time-series classification," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10*. Springer, 2017, pp. 495–511.

[92] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," *Computational Intelligence*, vol. 26, no. 3, pp. 232–257, 2010.

[93] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 155–164.

[94] Y. Zelenkov, "Example-dependent cost-sensitive adaptive boosting," *Expert Systems with Applications*, vol. 135, pp. 71–82, 2019.

[95] N. Günnemann and J. Pfeffer, "Cost matters: a new example-dependent cost-sensitive logistic regression model," in *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part I 21*. Springer, 2017, pp. 210–222.

[96] A. C. Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive logistic regression for credit scoring," in *2014 13th International conference on machine learning and applications*. IEEE, 2014, pp. 263–269.

[97] A. Iranmehr, H. Masnadi-Shirazi, and N. Vasconcelos, "Cost-sensitive support vector machines," *Neurocomputing*, vol. 343, pp. 50–64, 2019.

[98] L. Zhang and D. Zhang, "Evolutionary cost-sensitive extreme learning machine," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 12, pp. 3045–3060, 2016.

[99] P. Del Moral, S. Nowaczyk, and S. Pashami, "Why is multiclass classification hard?" *IEEE Access*, vol. 10, pp. 80 448–80 462, 2022.

[100] A. Fernández, V. López, M. Galar, M. J. Del Jesus, and F. Herrera, "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowledge-based systems*, vol. 42, pp. 97–110, 2013.

[101] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Advances in neural information processing systems*, vol. 10, 1997.

[102] M. Koziarski, M. Woźniak, and B. Krawczyk, "Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise," *Knowledge-Based Systems*, vol. 204, p. 106223, 2020.

[103] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," *Journal of Big data*, vol. 7, pp. 1–47, 2020.

[104] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang, "New imbalanced bearing fault diagnosis method based on sample-characteristic oversampling technique (scote) and multi-class ls-svm," *Applied Soft Computing*, vol. 101, p. 107043, 2021.

[105] L. Santos, F. N. Santos, P. M. Oliveira, and P. Shinde, "Deep learning applications in agriculture: A short review," in *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 1*. Springer, 2020, pp. 139–151.

[106] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Deep learning applications for covid-19," *Journal of big Data*, vol. 8, no. 1, pp. 1–54, 2021.

[107] A. Newaz, M. O. Faruque, R. Al Mahmud, R. H. Sagor, and M. Z. M. Khan, "Machine learning enabled multimode fiber specklegram sensors: A review," *IEEE Sensors Journal*, 2023.

[108] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of big data*, vol. 2, pp. 1–21, 2015.

[109] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *Ieee Access*, vol. 6, pp. 9375–9389, 2017.

[110] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with applications*, vol. 91, pp. 464–471, 2018.

[111] A. Ali-Gombe and E. Elyan, "Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network," *Neurocomputing*, vol. 361, pp. 212–221, 2019.

[112] A. Newaz, F. S. Haq, and N. Ahmed, "A case study on risk prediction in heart failure patients using random survival forest," in *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*. IEEE, 2021, pp. 1–6.

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

APPENDICES

A Appendix A Supplementary files

The supplementary files containing detailed dataset information, performance measures on individual datasets from different techniques, etc. are available in the following repository: https://github.com/newaz-aa

The codes to replicate the algorithms are also provided in the same repository.

List of Publications

Journals

- **Newaz, A.**, Mohosheu, M.S. and Al Noman, M.A., 2023. Predicting complications of myocardial infarction within several hours of hospitalization using data mining techniques. Informatics in Medicine Unlocked, 42, p.101361.
  DOI: https://doi.org/10.1016/j.imu.2023.101361.

Conference

- **Newaz, A.**, Mohosheu, M.S., Al Noman, M.A. and Jabid, T., 2024, April. iBRF: Improved Balanced Random Forest Classifier. In 2024 35th Conference of Open Innovations Association (FRUCT) (pp. 501-508). IEEE.
  DOI: https://doi.org/10.23919/FRUCT61870.2024.10516372.

- Mohosheu, M.S., al Noman, M.A., **Newaz, A.** and Jabid, T., 2024, May. A Comprehensive Evaluation of Sampling Techniques in Addressing Class Imbalance Across Diverse Datasets. In 2024 6th International Conference on Electrical Engineering and Information and Communication Technology (ICEEICT) (pp. 1008-1013). IEEE.
  DOI: https://doi.org/10.1109/ICEEICT62016.2024.10534464.

Under Review

- iCost: A Novel Instance Complexity Based Cost-Sensitive Learning Framework for Imbalanced Classification.

- A Unified Sampling Framework to Jointly Address Class Imbalance and Overlapping for Imbalanced Learning.