Modifying class distributions to improve the classification of minority group examples in a class-imbalanced dataset

Banele Mdluli[1] and Terence L. van Zyl[2]

University of Johannesburg, Johannesburg, JHB, RSA

Abstract. Class-imbalanced datasets are a common occurrence in real-world applications. The imbalance between minority and majority classes exists due to the over-representation of one class compared to another in a dataset. The class imbalance might reflect a system's behaviour over time. However, the class imbalance causes sub-optimal performance for machine learning models that predict the system's future behaviour. Various techniques are used to reduce the negative impact of class-imbalanced datasets on machine learning models. Data resampling techniques are one of the main techniques, and the subdivisions of data resampling techniques include oversampling and undersampling. Oversampling techniques have outperformed undersampling techniques in most studies, and most data resampling techniques are derived from oversampling. However, some oversampling techniques are ineffective when used on minority-class datasets that lack within-class variation and have a high-class imbalance. In this study, an analysis was performed to understand the changes in within-class variation before and after oversampling for nine datasets. Additionally, classification performance was measured for standard and hybrid oversampled datasets. A novel hybrid oversampling technique that uses k-Means and ADASYN was implemented. Hybrid oversampling techniques generated synthetic examples that marginally changed the within-class variation and had the highest F1 score compared to standard oversampling techniques across nine datasets.

Keywords: Class imbalance · Oversampling · ADASYN · Classification algorithm · Within-class variation.

1 Introduction

A class-imbalanced dataset occurs when one class (the majority class) has significantly more examples than the other (the minority class) [22]. Such an imbalance may cause model prediction errors because a model trained on imbalanced data tends to misclassify examples belonging to the minority class [17]. Certain applications of classification models focus on predicting the minority class outcomes (e.g. machine failure, credit fraud detection, early cancer detection) [13, 4, 28]. In those scenarios, class-imbalanced datasets pose a significant threat to model minority class prediction.

Class imbalanced datasets are common in real-world applications [12]. The imbalance in a dataset may result from the type of real-world application (e.g., fraud detection systems) or the expense of observing rare undesired events (e.g., space shuttle oil spillage) [22, 17]. Class-imbalanced datasets affect machine learning model predictions in varying ways. The most common negative impact is misclassifying examples from the minority class as the majority class. Misclassifying minority class examples might lead to an increase in False Negative(FN) outcomes.

In some cases, incorrectly predicting an outcome, particularly a false positive (FP) outcome, is associated with low costs/consequences [23, 5]. However, predicting a False Negative (FN) outcome can lead to significant costs or consequences, particularly in cases involving class-imbalanced datasets [23]. To reduce the cost of ML models incorrectly predicting outcomes, precautions are taken when imbalanced datasets are used for model predictions, since rare events are hard to learn for most machine learning models[6].

Moreover, the impact of class-imbalanced datasets extends beyond just model predictions; it also raises ethical considerations and fairness issues in machine learning applications [5]. Models trained on class-imbalanced datasets may reinforce biases observed in datasets, potentially leading to unfair or discriminatory outcomes, especially when predicting underrepresented classes [12]. These biases can have far-reaching consequences, negatively affecting decision-making concerning lending [26], hiring [10], and criminal execution [1].

Classifying minority class examples in a class-imbalanced dataset has become a rigorously researched topic. Highly imbalanced datasets often result in classification models with high accuracy scores, which is not a reliable performance indicator [22]. Various data underlying properties affect the classification of minority classes. The most prevalent underlying properties are the training size of the dataset, the complexity of the dataset, and the degree of imbalance [16]. Multiple techniques have been proposed to reduce the most common issue about class-imbalanced datasets, which is misclassifying minority class examples [17]. However, most techniques focus on improving the classification of minority-class examples without addressing the underlying dataset properties that contribute to the misclassification of minority-class examples.

2 Problem statement

Addressing class imbalance in datasets is a common challenge in machine learning. Existing approaches primarily focus on correcting the imbalance between minority and majority classes without considering the impact on the dataset's intrinsic properties. Some datasets are inherently imbalanced due to the nature of their application, e.g., space shuttle oil spillage data[12]. Rectifying the imbalance might remove essential information or introduce noise in the dataset that might hinder model performance [22].

A key issue in class-imbalanced datasets is the lack of within-class variation, particularly in the minority class, which often has fewer distinct examples [17,

21]. The problem occurs when resampling techniques create synthetic examples with minimal variation, worsening the misclassification of minority class examples [17]. This problem is further compounded by minority and majority class examples that are close in similarity, making it difficult for models to distinguish between them accurately [3].

Most existing methods for handling class imbalance focus primarily on improving the classification of minority class examples while neglecting the issue of within-class variation. However, both challenges often coexist in a single dataset, impacting the model's ability to learn meaningful patterns. This study aims to analyse and address both factors using a novel technique to improve the model's output trained on class-imbalanced datasets.

3 Background

The techniques for class-imbalanced datasets consist of 3 categories. The first category is data resampling techniques, which attempt to reduce the imbalance level between the majority and minority classes during the data pre-processing phase of a machine learning model [17]. Algorithm techniques modify the underlying Machine learning model parameters or hyperparameters to mitigate biases in favour of the majority class [22, 23]. Hybrid techniques utilise data resampling and algorithm techniques to mitigate issues concerning class-imbalanced datasets [17]. The background section briefly discusses Data resampling, Algorithm-based, hybrid techniques and related work.

3.1 Algorithm Methods

Cost-sensitive learning is one of the Algorithm-based techniques used to counteract the negative effects of class-imbalanced datasets. Elkan [11] first introduced the method in 2001, demonstrating that a cost matrix could prioritise the minority class and reduce misclassification. Traditional classification algorithms like Naive Bayes, Artificial Neural Networks and Support Vector Machines aim to minimize the overall error rate [23]. Reducing incorrectly predicted outcomes, regardless of the class, ignores the differences in misclassification errors and assumes that all errors are identical [23]. In some real-world applications, the assumption is false. Therefore, cost-sensitive learning aims to minimise a misclassification error that is a costly outcome [23, 11].
The limitation of Cost-Sensitive learning is that no standard approach is used to determine the cost values. Sometimes, a domain expert is required to assign the cost values [11, 8].

3.2 Hybrid techniques

Traditional machine-learning algorithms may struggle to obtain desired results when datasets are highly imbalanced [9]. The inability to capture or learn essential underlying patterns from class-imbalanced datasets makes it challenging for

traditional machine learning models to function optimally [9]. Ensemble learning algorithms attempt to tackle the limitations of traditional machine learning models when using highly imbalanced datasets. Ensemble learning algorithms aim to learn various patterns on a dataset by extracting subsets and training multiple learners (models) based on those subsets [9]. Thereafter, a voting or weighing mechanism is imposed on all learners to create an aggregate model that is better than one learner [27].
Ensemble learning algorithms attempt to reach a balance point between bias and variance using Boosting and Bagging. Bagging is a shortened term for Bootstrap Aggregating; the approach reduces high variances, which can lead to overfitting, especially in high variance models like Decision Tree [27].
The boosting approach trains models sequentially, where the subsequent model focuses on rectifying the previous model's error to improve overall accuracy [25].

3.3 Data Resampling Techniques

There are multiple data under-sampling methods used to balance the distribution between majority and minority classes. Under-sampling techniques remove majority class examples to balance or reduce the imbalance between classes [12]. There are two types of under-sampling methods: non-heuristics and heuristics sampling methods. Non-heuristic techniques randomly remove examples from the majority class to reduce the imbalance ratio between majority and minority groups; these methods are known as Random Under-Sampling techniques (RUS) [19]. These methods are considered naïve since they assume that most events resulting from a specific feature are independent of another feature. This assumption in real-world applications is usually incorrect [19]. Non-heuristic techniques do not improve model performance significantly when used with traditional classification models. However, their application in deep learning has produced better results compared to heuristic methods [17].
On the other hand, heuristic techniques focus on retaining vital information that can help a model learn about various classes while removing redundant examples. [12]. Multiple heuristic undersampling methods aim to strategically remove certain majority class examples based on their significance.

There are multiple data oversampling-sampling techniques used to balance the distribution between majority and minority classes. Data oversampling methods adjust the dataset's distribution and decrease the imbalance between the majority and the minority class by adding minority class examples [17]. Oversampling techniques are also divided into two categories, which are heuristics and non-heuristics.

Non-heuristic techniques randomly duplicate examples of the minority class to balance the distribution between the majority and minority classes. These methods are known as Random Over-Sampling Techniques (ROS) [12]. The challenge with ROS methods is that they tend to increase the likelihood of over-fitting. Over-fitting occurs due to generating multiple copies of the same minority class examples in a dataset [20]. ROS methods are known to underperform when used with traditional classification models. However, modern deep

learning applications have seen ROS outperform other alternative methods [17]. Heuristic oversampling techniques attempt to overcome the challenges of ROS techniques. Heuristic oversampling techniques reduce over-fitting and class imbalance using interpolation to generate synthetic examples [12]. Many of these heuristic methods for oversampling are derived from SMOTE (Synthetic Minority Over-sampling Technique) [17]. SMOTE generates synthetic examples that are similar to the original class examples but differ slightly to add variety [4]. SMOTE performs better than ROS methods in cases where the class imbalance is high [4]. The increased popularity of SMOTE led to different variations of SMOTE.

Coefficient of Variation for Measuring Within-class variations

Class imbalance is a common challenge in machine learning, where the distribution of classes in a dataset is highly skewed, with one class significantly outnumbering the others [22, 17]. This issue can lead to biased model performance and decreased accuracy, specifically for the minority class. Past research papers have explored various techniques to mitigate the impact of class imbalance, and one crucial aspect is assessing the within-class variation [2, 24]. The coefficient of variation (CV) is a widely used statistical measure that relates the standard deviation to the mean of a dataset [15, 18]. It is particularly valuable when analysing datasets with different scales or units. In the context of imbalanced datasets, the CV has emerged as a valuable tool for measuring within-class variation [2]:

$$CV = \frac{\sigma}{\mu} \tag{1}$$

where σ is the standard deviation of a set and μ is the mean of a set. One of the primary reasons for using CV in imbalanced datasets is its ability to normalise the variation across features or classes. In class imbalance datasets, the minority class may exhibit higher or lower variability due to its smaller sample size [16]. This variability can adversely affect model training and generalisation [23, 16]. By calculating the CV within each class, researchers can obtain a standardised measure of variation that facilitates fair class comparisons [7].

4 Related work

SMOTE generates synthetic examples that are similar to the original class examples but differ slightly to add variety [4]. SMOTE performs better than ROS methods in cases where the class imbalance is high [4]. The increased popularity of SMOTE led to different variations of SMOTE. In some instances, the borderline class examples are essential for classification. Borderline SMOTE was developed to generate minority examples close to the borderline [13]. In 2005, Han, Wang and Mao conducted an experiment using a linearly inseparable dataset

[13]. The investigation showed that borderline-SMOTE achieved a better True Positive rate and F1-Score than ROS and SMOTE for the minority group [13].
In 2009, Bunkhumpornpat, Sinapiromsaran and Lursinsap [3] proposed a method called Safe level-SMOTE, which assigns each positive instance a safe level value before generating synthetic examples. A safe level value is the number of positive samples in k-NN [3]. The safe level ratio is used to categorise the state of a positive instance into noise or safe. If a safe level is close to 0, the positive instance is considered noise, but if it is close to 1, it is considered safe. An experiment by the same authors using two datasets from the UCI repository shows that safe-level SMOTE outperforms SMOTE and Borderline SMOTE.
In 2008, Haibo and Yang proposed a technique called Adaptive Synthetic Oversampling (ADASYN) [14], based on generating synthetic minority class examples using a minority class distribution. The technique focuses on generating minority class examples that are harder for the classifier to learn than those that are not. It was observed that the method outperforms the SMOTE technique in experiments performed by Haibo and Yang [14].
Under-sampling and Over-sampling techniques help address the challenges of imbalanced datasets during data pre-processing. However, there is a limitation regarding improving the within-class variation and reducing overlapping class distribution. Most methods focus on one of the two factors. Therefore, considering both factors might provide a robust approach that this dissertation analysed.

5 Dataset

Nine datasets were used for the research study. The datasets are available on the Kaggle repository. Every dataset contains a target variable that has Boolean values (e.g., True or False, 1 or 0). The names of the datasets are credit-card-fraud-detection-dataset-2023, telco-customer-churn, stock-data-with-industry-information, heart-failure-prediction, loan-default and social-network-ads. All datasets had a class imbalance, where the minority class represented at most 15% of the dataset. Each dataset had at a minimum 2000 records (instances). The least number of independent variables was eight for a dataset. Multidisciplinary datasets were used to investigate each resampling method.

6 Resampling techniques

In this study, the resampling techniques were selected based on their resemblance to the proposed technique and the resampling type. SMOTE was selected since it is the base technique, from which most techniques are derived. k-SMOTE is a hybrid technique that is similar to the proposed technique (k-ADASYN). ADASYN constitutes one component of the proposed technique, which integrates both k-Means and ADASYN. There are numerous other techniques that could have been considered; however, the ones selected had the most resemblance based on the literature.

Experiment 1 Aimed to evaluate if the hybrid or standard oversampling techniques overestimate or underestimate the coefficient of variation for different minority class subsets. The experiment aimed to measure each oversampling technique's accuracy in estimating the expected Coefficient of Variation (CV) for a minority class subset.

1. A subset was created by randomly selecting R% of the minority class examples.
2. The hybrid and standard oversampling techniques were used to increase the minority class subset by 100%.
3. The coefficient of variation was calculated for the oversampled subset.
4. The expected coefficient of variation was calculated using Bootstrap, and the following sub-procedures were performed:
   (a) A subset was created by randomly selecting minority class examples with replacements. The subset created was $2 \times$ R% of the minority class examples.
   (b) The coefficient of variation was calculated for the subset and recorded.
   (c) Steps 4.a to 4.b were repeated 1000 times using different samples (bootstrap samples).
   (d) The expected CV was calculated using an average of all 1000 bootstrap sample CVs.
5. Each oversampling technique's CV values were compared to the bootstrap expected CV value.
6. The CV values of the standard oversampling technique were compared to those of the hybrid oversampling technique.

Experiment 2 Aimed to measure and compare ML classification performance scores before and after using standard and hybrid oversampled datasets.

1. SMOTE, ADASYN, k-SMOTE, and K-ADASYN [1] were performed on the minority class to achieve an imbalance ratio of 70:30 (majority: minority class).
2. The Decision Tree, XGBoost and Random Forest models were used to evaluate the prediction performance of the oversampled dataset for each oversampling technique.
3. The hyperparameters of each classification model were tuned to enhance overall classification performance based on the oversampled dataset.
4. The F1 score and AUC for each classification model were computed to evaluate the efficacy of each oversampling technique.
5. The performance scores of the standard oversampling techniques were compared to those of the hybrid oversampling techniques.
6. The performance scores for hybrid oversampling techniques were compared between the two hybrid techniques.

[1] k-ADASYN was the proposed new technique for implementing a hybrid oversampling technique

7 Experiment Results

7.1 Experiment 1

Table 1. Coefficient of variation for oversampled and bootstrapped data subsets, the subsets were 25%, 35%, and 45% of the minority class examples across all datasets.

Dataset	Minority class subset size	Bootstrap	SMOTE	ADASYN	k-SMOTE	k-ADASYN
1	25%	**1.77**	1.67	1.67	1.68	**1.81**
	35%	**1.78**	1.68	1.64	1.66	**1.79**
	45%	**1.78**	1.60	1.63	1.59	**1.75**
2	25%	**2.95**	3.19	**2.95**	2.99	**2.95**
	35%	**2.92**	**2.56**	2.40	2.49	**2.56**
	45%	**2.93**	**2.88**	2.33	2.50	2.62
3	25%	-4.01	-4.01	-4.01	-4.01	-4.01
	35%	**-4.02**	-3.04	-3.10	-2.69	**-3.11**
	45%	**-4.02**	-2.96	-3.08	-2.87	**-3.24**
4	25%	**3.34**	3.10	3.23	3.56	**3.34**
	35%	**3.33**	3.13	3.58	**3.24**	3.47
	45%	**3.30**	**3.65**	3.69	4.05	4.10
5	25%	**2.82**	2.93	**2.81**	2.75	2.91
	35%	**2.82**	1.92	2.04	2.03	**2.35**
	45%	**2.82**	1.86	2.10	2.23	**2.35**
6	25%	-4.67	-4.67	-4.67	-4.67	-4.67
	35%	**-4.66**	-4.39	**-4.73**	-3.84	-4.87
	45%	**-4.72**	**-4.49**	-5.71	-3.60	-5.40
7	25%	-8.04	-8.04	-8.04	-8.04	-8.04
	35%	**-7.95**	-8.45	-6.72	-9.13	**-8.26**
	45%	**-7.99**	**-7.84**	-6.65	-7.59	-7.59
8	25%	1.37	1.37	1.37	1.37	1.37
	35%	**1.37**	1.18	1.21	1.34	**1.39**
	45%	**1.35**	1.13	1.17	1.09	**1.27**
9	25%	**25.77**	29.97	22.17	50.89	**25.77**
	35%	**26.58**	9.04	6.86	8.96	**10.73**
	45%	**28.78**	14.60	9.02	**21.97**	16.04

Table 1 shows that when the expected bootstrap dataset CV was low, it marginally differed from the oversampled dataset CV. The lowest difference between the expected bootstrapped and oversampled dataset CV was observed in Datasets 1, 2, and 8, where the expected CV was low compared to other datasets. When the expected bootstrap dataset CV was high, it significantly differed from the oversampled dataset CV. The highest difference between the expected and oversampled dataset CV was observed in Dataset 8, where the expected CV was the highest compared to other datasets. The coefficient of variation (CV) values for the Bootstrap and Oversampling techniques that were identical or closely similar were highlighted in bold to indicate their strong correlation. Where CV values were identical for all techniques, they were not highlighted in bold.

In most instances where the expected Bootstrap CV was low, the oversampled dataset CV was identical to the expected bootstrapped CV for the 25% minority class subset size. When the minority class subset size increased to 35% and

45%, the expected bootstrapped dataset CV marginally differed from the oversampled dataset CV (e.g., Dataset 1, 2, 8). When the expected bootstrapped dataset CV was high, the oversampled dataset CV significantly differed from the bootstrapped dataset CV (e.g., Datasets 4, 7, 9). Based on results shown in Table 1, oversampling techniques overestimate and underestimate the expected coefficient of variation. In most instances, Hybrid Oversampling techniques overestimate the expected bootstrapped dataset CV, while Standard Oversampling techniques underestimate it (e.g., Datasets 1, 2, 8, 9). However, there are instances where all oversampling techniques overestimate or underestimate the expected CV. Therefore, the CV overestimation and underestimation depend on the dataset's properties.

Increasing (overestimating) the CV of the dataset improves the within-class variation by introducing dispersion or variability in the class. Improving the within-class variation for the minority class was performed to assist machine learning models in correctly classifying minority class examples. Three machine learning models were used to check if oversampling techniques used to modify the within-class variation improved the classification of minority class examples.

7.2 Experiment 2

The objective of experiment 2 was to verify whether modifying the distribution of the minority class improves the classification of minority class examples. To prevent data leakage, the test data was scaled using the mean and standard deviation of the training dataset. The method which was used to scale the data was Standardisation. The classification models were trained using 70% of the dataset for each of the nine datasets. The models were tested using the remaining 30% that was not used to train each model. The following table shows the classification performance metric scores for the original and oversampled datasets.

Extreme Gradient Boosting (XGBoost) The Extreme Gradient Boosting (XGBoost) tuned hyperparameters were learning rate, max depth, number of estimators, sub-sample and colsample bytree. The following table shows the classification performance metric scores for the original and oversampled datasets.

Table 2 shows the XGBoost classification results for all nine datasets. The classification performance scores shown were f1 and AUC. The standard techniques outperformed hybrid techniques (e.g., Datasets 2,3,6,7,9). However, the k-ADASYN f1 score and AUC show that the precision and accuracy scores for the XGBoost algorithm were lowest for most datasets.

Random Forest Model Results The Random Forest Model tuned hyperparameters were max depth, minimum sample leaf, minimum sample split, max features and criterion. The following table shows the classification performance metric scores (F1 and AUC) for the original and oversampled datasets.

Table 2. XGB Classification model performance result for all nine datasets

Dataset no.	Performance Score	Original	ADASYN	k-SMOTE	SMOTE	k-ADASYN
1	F1 score	**99.99**	99.98	99.98	**99.99**	**99.99**
	AUC Score	**100.00**	99.99	99.99	99.99	99.99
2	F1 score	99.62	**100.0**	**100.0**	99.68	99.63
	AUC Score	99.62	**100.0**	**100.0**	99.68	99.63
3	F1 score	50.91	**55.44**	52.84	52.17	55.08
	AUC Score	66.80	**70.29**	68.77	68.38	68.99
4	F1 score	100.0	100.0	100.0	100.0	100.0
	AUC Score	100.0	100.0	100.0	100.0	100.0
5	F1 score	10.13	78.98	80.00	77.09	**87.82**
	AUC Score	52.85	85.20	84.85	82.26	**90.71**
6	F1 score	81.33	80.14	**81.43**	79.27	79.63
	AUC Score	80.79	84.61	**85.56**	83.97	75.12
7	F1 score	**43.69**	42.89	35.31	34.87	41.67
	AUC Score	61.63	**63.76**	60.56	60.41	60.76
8	F1 score	72.22	64.20	65.82	64.20	**79.45**
	AUC Score	79.71	77.20	77.77	77.20	**83.47**
9	F1 score	71.01	78.61	**78.87**	76.61	65.83
	AUC Score	78.80	83.09	**83.32**	82.13	75.57

Table 3. Random Forest Classification model performance result for all nine datasets

Dataset no.	Performance Score	Original	ADASYN	k-SMOTE	SMOTE	k-ADASYN
1	F1 score	99.99	99.98	99.99	99.99	99.98
	AUC Score	99.99	99.99	99.99	99.99	99.98
2	F1 score	99.62	100.0	100.0	99.68	100.0
	AUC Score	99.62	100.0	100.0	99.68	100.0
3	F1 score	48.16	56.84	56.84	56.84	50.21
	AUC Score	65.11	70.94	70.94	70.94	66.28
4	F1 score	100.0	100.0	100.0	100.0	100.0
	AUC Score	100.0	100.0	100.0	100.0	100.0
5	F1 score	0.00	79.93	82.96	77.67	**87.43**
	AUC Score	50.00	85.86	87.82	83.94	**91.15**
6	F1 score	**83.17**	80.00	80.00	80.00	82.69
	AUC Score	82.46	84.37	84.37	84.37	80.10
7	F1 score	**41.74**	34.67	34.67	34.67	39.47
	AUC Score	**61.53**	60.35	60.35	60.35	60.66
8	F1 score	77.33	64.20	64.20	64.20	**80.00**
	AUC Score	83.77	77.20	77.20	77.20	**85.33**
9	F1 score	67.45	72.49	77.13	**78.31**	70.65
	AUC Score	75.72	78.80	81.78	**82.51**	78.29

Table 3 shows the Random Forest classification results for all nine datasets. The classification performance scores shown were F1 and AUC. Overall, the standard techniques performed the same as the hybrid techniques. In some cases, the ADASYN, SMOTE and k-SMOTE had the same F1 score (e.g., Datasets

4, 2, 1). k-ADASYN was an outlier when compared to the other techniques for datasets 5 and 8. The difference between the hybrid and standard oversampling technique's classification scores was small since some scores were the same for both hybrid and standard techniques. Overall, the classification of minority class examples improved after oversampling using hybrid and standard techniques. The F1 score was higher for hybrid techniques than standard ones across all three classification models. While the AUC scores varied based on the dataset.

Decision Tree Model Results The Extreme Gradient Boosting (XGBoost) tuned hyperparameters were n_estimators, criterion, max_depth and max_features. The following table shows the classification performance metric scores for the original and oversampled datasets.

Table 4. Decision Tree Classification model performance result for all nine datasets

Dataset no.	Performance Score	Original	ADASYN	k-SMOTE	SMOTE	k-ADASYN
1	F1	99.20	99.61	99.81	98.94	**99.99**
	AUC	99.64	99.78	99.88	99.32	**99.99**
2	F1 score	95.82	94.46	98.56	98.73	**99.63**
	AUC Score	97.60	98.08	99.39	99.43	**99.63**
3	F1 score	51.20	43.99	47.38	43.99	**55.08**
	AUC Score	66.98	64.32	66.10	64.32	**68.99**
4	F1 score	98.35	91.76	99.65	97.86	**100.0**
	AUC Score	99.67	92.79	99.92	98.16	**100.0**
5	F1 score	0.00	57.81	65.09	57.70	**87.82**
	AUC Score	49.89	73.49	76.74	73.12	**90.71**
6	F1 score	**79.86**	63.53	71.16	63.53	79.63
	AUC Score	**80.14**	74.39	79.02	74.39	75.12
7	F1 score	**42.04**	36.39	39.87	36.39	41.67
	AUC Score	59.84	60.64	**62.44**	60.64	60.76
8	F1 score	77.33	65.12	54.55	65.12	**79.45**
	AUC Score	83.77	78.71	70.95	78.71	**83.47**
9	F1 score	56.95	63.06	**67.48**	66.00	65.83
	AUC Score	72.60	75.21	**77.86**	76.82	75.57

Table 4 shows the Decision Tree classification results for all nine datasets. The classification performance scores shown were F1 and AUC. The hybrid techniques outperformed standard techniques (e.g., Datasets 1, 2, 3, 4, 5, 8). The k-ADASYN outperformed k-SMOTE for most of the datasets used; only a few datasets (e.g., Dataset 9), where k-SMOTE outperformed k-ADASYN. The oversampling of the minority class improved the classification of minority-class examples for most datasets (e.g., Datasets 1, 2, 3, 4, 5, 6, 9).

8 Result Discussion

Oversampling the minority class affects the within-class variation (coefficient of variation) in various ways. Based on the coefficient of variation, the changes in the within-class variation differ from one dataset to the other. However, the common observation across all datasets was that the coefficient of variation of the minority class dataset and the oversampled minority class dataset was the same or almost the same when the CV was small (close to zero). When the minority class dataset CV was large (greater than 10), the oversampled minority class CV was significantly lower than the original minority class dataset. When the oversampled dataset's CV was similar to the original dataset's CV, it indicated that the synthetic examples generated through oversampling marginally improved the within-class variation. When the oversampled dataset CV was higher than the original dataset CV, the generated synthetic examples improved the within-class variation.

Hybrid Oversampling Techniques generated a CV value more similar to the original minority class CV than Standard Oversampling Techniques. The similarities in CVs indicate that Hybrid Oversampling Techniques generated synthetic examples that closely resemble the original data. A high CV similarity between the synthetic and the original examples reduces the likelihood of misclassifying minority class examples. The low dispersion or variability ensures that minority classes are easily identifiable based on their similar characteristics. However, unknown distinct minority class examples might be misclassified if they were not in the training set. A low CV similarity between the synthetic and original dataset examples might have a higher likelihood of misclassifying the minority class examples. However, unknown distinct minority class examples might be correctly classified due to the synthetic examples containing more distinct characteristics than the original dataset.

The Hybrid Oversampling Techniques had the highest average F1 score across all nine datasets and had the most similar minority class dataset CV when compared to the original minority class dataset CV. The High F1 score was observed when the model's recall score was higher than the precision score. Standard Oversampling Techniques had the highest F1 score when the precision score was higher than the recall score. Therefore, choosing Hybrid or Standard Oversampling Techniques depends on the importance of the recall or precision score.

9 Conclusion

The classification of the oversampled datasets outperformed the original datasets when comparing performance metrics. However, when the initial coefficient of variation was very low, the original dataset classification performance was nearly identical to the oversampled dataset classification performance. When the initial coefficient of variation was high, classification scores for oversampling techniques significantly outperformed the original dataset classification score. The

hybrid oversampled dataset classification scores were higher than the standard ones for the Decision Tree algorithm and differed marginally for XGBoost and Random Forest. k-ADASYN classification scores outperformed all other oversampling techniques using the Decision Tree algorithm. Based on the F1 score, where the recall score is very high, the proposed novel hybrid technique should be used in cases where predicting a false negative is more costly than predicting a false positive.

10 Recommendations

There is little research on hybrid oversampling or undersampling techniques. The combination of clustering and data resampling techniques might offer tremendous performance gain. Therefore, other clustering-based oversampling or undersampling techniques should be explored to understand their performance against other standard techniques. Additionally, the k-ADASYN and Decision Tree algorithms should be used with various datasets to test whether similar observations can be obtained in another study.

Disclosure of Interests. The author declares no competing interests relevant to the content of this article.

References

1. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias. In: Ethics of data and analytics, pp. 254–264. Auerbach Publications (2022)
2. Bindu, K.H., Morusupalli, R., Dey, N., Rao, C.R.: Coefficient of variation and machine learning applications. CRC Press (2019)
3. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: 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. pp. 475–482. Springer (2009)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
5. Devi, D., Biswas, S.K., Purkayastha, B.: A review on solution to class imbalance problem: undersampling approaches. In: 2020 international conference on computational performance evaluation (ComPE). pp. 626–631. IEEE (2020)
6. Devi, D., Biswas, S.K., Purkayastha, B.: A review on solution to class imbalance problem: undersampling approaches. In: 2020 international conference on computational performance evaluation (ComPE). pp. 626–631. IEEE (2020)
7. Ding, X., Chu, J., Hu, J., Yu, H., Li, T.: A deep unsupervised representation learning architecture using coefficient of variation in hidden layers. In: 2023 IEEE 9th International Conference on Cloud Computing and Intelligent Systems (CCIS). pp. 182–186. IEEE (2023)

8. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 155–164 (1999)
9. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Frontiers of Computer Science **14**, 241–258 (2020)
10. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference. pp. 214–226 (2012)
11. Elkan, C.: The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence. vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
12. Guo, X., Yin, Y., Dong, C., Yang, G., Zhou, G.: On the class imbalance problem. In: 2008 Fourth international conference on natural computation. vol. 4, pp. 192–201. IEEE (2008)
13. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1. pp. 878–887. Springer (2005)
14. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). pp. 1322–1328. IEEE (2008)
15. Hendricks, W.A., Robey, K.W.: The sampling distribution of the coefficient of variation. The Annals of Mathematical Statistics **7**(3), 129–132 (1936)
16. Japkowicz, N.: Learning from imbalanced data sets: a comparison of various strategies. In: AAAI workshop on learning from imbalanced data sets. vol. 68, pp. 10–15. AAAI Press Menlo Park, CA (2000)
17. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. Journal of Big Data **6**(1), 1–54 (2019)
18. Kesteven, G.L.: The coefficient of variation. Nature **158**(4015), 520–521 (1946)
19. Kotsiantis, S.B., Pintelas, P.E.: Mixture of expert agents for handling imbalanced data sets. Annals of Mathematics, Computing Teleinformatics **1**(1), 46–55 (2003)
20. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: Icml. vol. 97, p. 179. Citeseer (1997)
21. Lee, H., Park, M., Kim, J.: Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In: 2016 IEEE international conference on image processing (ICIP). pp. 3713–3717. IEEE (2016)
22. Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A., Seliya, N.: A survey on addressing high-class imbalance in big data. Journal of Big Data **5**(1), 1–30 (2018)
23. Ling, C.X., Sheng, V.S.: Cost-sensitive learning and the class imbalance problem. Encyclopedia of machine learning **2011**, 231–235 (2008)
24. Liu, T., Qu, S., Zhang, K.: A clustering algorithm for automatically determining the number of clusters based on coefficient of variation. In: Proceedings of the 2nd International Conference on Big Data Research. pp. 100–106 (2018)
25. Mayr, A., Binder, H., Gefeller, O., Schmid, M.: The evolution of boosting algorithms. Methods of information in medicine **53**(06), 419–427 (2014)
26. Mukerjee, A., Biswas, R., Deb, K., Mathur, A.P.: Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management. International Transactions in operational research **9**(5), 583–597 (2002)
27. Suthaharan, S.: Machine learning models and algorithms for big data classification. Integr. Ser. Inf. Syst **36**, 1–12 (2016)

28. TOMEK, I.: Two modifications of cnn. IEEE Trans. Systems, Man and Cybernetics **6**, 769–772 (1976)