Earthline Journal of Mathematical Sciences
E-ISSN: 2581-8147
Volume 16, Number 2, March 2026, Pages 221-235
https://doi.org/10.34198/ejms.16226.17.221235

A Dual Sampling Approach for Improved Classifier Performance on Imbalance Datasets

Moses Apambila Agebure[1,*], Abdul-Wakil Yakubu Iddrisu[2],
Gabrial Armah[3] and Stephen Akobre[4]

[1] Department of Computer Science, School of Computing and Information Sciences,
C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana
ORCID: https://orcid.org/0000-0003-3555-8349
e-mail: magebure@cktutas.edu.gh

[2] Department of Computer Science, School of Computing and Information Sciences,
C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana
ORCID: https://orcid.org/0009-0001-8163-2637
e-mail: awiddrisu.stu@cktutas.edu.gh

[3] Department of Business Computing, C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana
e-mail: garmah@cktutas.edu.gh

[4] Department of Cyber Security and Computer Engineering Technology,
C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana
ORCID: https://orcid.org/0000-0003-3320-212X
e-mail: sakobre@cktutas.edu.gh

Abstract

Background: The inability of traditional machine learning models to adequately classify minority instances in imbalanced datasets is a known challenge that militate against the successful application of these models in several real-world domains. To address this problem, several techniques including data sampling are mostly used. Though reducing the imbalance ratio via sampling is reported to improve classifier performance, most approaches do not consider the intra-class distribution of instances while sampling, which often lead to loss of significant information or on the contrary cause data redundancy. **Methods:** This study proposes a novel Dual Sampling Technique (DST) that minimises these challenges and enhances classifier performance on imbalance datasets. The technique proceeds by first clustering a training set into a number of clusters determined a priori using the elbow method. Sampling ratios are computed from each cluster and either random undersampling or a novel average oversampling technique or both are used to perform sampling in each cluster depending on the imbalance ratio. The resulting datasets are used to train Random Forest, Decision Tree and K-Nearest Neighbor classifiers and their performance evaluated. **Findings:** Experimental results showed that the performance of the classifiers significantly improved in most cases when the proposed technique is used to sample the training set prior to model building than when Random Undersampling

Received: December 11, 2025; Accepted: January 6, 2026; Published online: February 5, 2026
2020 Mathematics Subject Classification: 68T05.
Keywords and phrases: machine learning, oversampling, undersampling, class imbalance learning, classification algorithm, clustering.

[*]Corresponding author

(RUS), Random Oversampling (ROS), Synthetic Minority Oversampling Technique (SMOTE) and Cluster-Based Undersampling (CBU) are used. **Novelty:** The novelty of the proposed technique lies in the exploration of a unique concept that sought to minimise the imbalance ratio in datasets while maintaining their natural distribution by uniquely performing both undersampling and oversampling on the same dataset.

1 Introduction

In fields such as computer science, biology, economics, among others, machine learning has become an innovative tool for solving a range of real-life problems [1]. Despite the rising importance of machine learning in solving these problems, there are a number of factors that affect the learning abilities of algorithms of which class imbalance is an important example. Class imbalance is a phenomenon where instances in one class of a dataset are overrepresented compared to that in other classes [2]. Datasets from numerous real-world application areas, such as fraud detection, medical diagnosis, natural language processing and others are often confronted with this problem [3, 4].

The issue of class imbalance is widely acknowledged and has drawn increasing attention from both academia and industry over the past two decades. Attempts have been made to address this problem using various undersampling and oversampling strategies that reduce or increase the number of instances in the majority or minority classes, respectively. This is often done with the aim of obtaining datasets with relatively balanced number of instances in each class [5]. Despite making progress in this regard, researchers are still developing improved techniques to efficiently handle the class imbalance problem particularly, in datasets with other peculiar issues such as noise and outliers in order to retain the most relevant instances for learning.

In some cases, class imbalance may co-occur with a situation that can best be described as the within class imbalance problem [6]. Within class imbalance may arise due to data scarcity in which certain sub-concepts within a class are under-represented. For instance, in a fraud detection dataset, fraudulent class instances may comprise of different types of fraudulent activities, some of which may exhibit rare occurrence while others may be prevalent leading to a within class imbalance situation. Consequently, learning to correctly classify these rare sub-concepts within a minority class pose serious challenges [7].

As mentioned, the occurrence of class imbalance is intrinsic in most real-world domains and cannot be avoided in any way [5]. Thus, it requires continuous effort towards developing efficient techniques to mitigate its effects. To this end, several approaches for minimising the effects of imbalance datasets have been proposed over the years, which can be categorised mainly into data level, algorithmic level and hybrid techniques [8].

Data level techniques are commonly categorised into two; oversampling and undersampling [9]. While undersampling is selecting subsamples from the majority class, oversampling includes replicating or creating artificial samples of the minority class instances [10, 11, 12, 13]. Sampling is appealing since it simply requires modifying the size of the training data and not the learning algorithm. Data-level approaches are popular and effective in addressing the class imbalance problem [14].

However, undersampling techniques, particularly, random undersampling, has the inherent disadvantage of potentially discarding important majority class instances from the training dataset. To overcome this, there have been significant attempts toward developing seemingly intelligent undersampling techniques that attempt to minimise the trade-off between balancing the data distribution and maintaining the most important instances for training. These approaches often employ advanced computational schemes such as optimisation/meta-heuristic-based techniques [15], clustering techniques [16], margin theory [17], among others. For instance, [5] proposed an undersampling technique that leverages clustering and event selection techniques. They employed the k-means algorithm to cluster instances in the majority class while using a cluster weight metric to determine the number of instances that should be selected from a cluster. The Mahalanobis distance of each instance in a cluster to the centroid is computer and used as a guide to select the most representative samples from the clusters. This they hope would ensure that the selected instances reserve the distribution of instances in the cluster. Similarly, [18] introduced a clustering-based undersampling technique to improve the performance of the C4.5 classification algorithm. They employed the Clustering Large Application (CLARA) algorithm to address class imbalance while ensuring that valuable samples are not discarded during the process.

Random oversampling (ROS) on the other hand, is reportedto mostly lead to data redundancy resulting to classifier over-fitting and longer training time. One of the foremost and widely used technique that was introduced to minimise the limitations of random oversampling is the Synthetic Minority Oversampling Technique (SMOTE) [10]. However, as highlighted in [8], the SMOTE has similar limitations to the random oversampling technique. As such, several variants of the SMOTE and other enhanced approaches have been introduced to overcome these inherent limitations and improve the general performance of classifiers. For example, [19] proposed an oversampling technique to deal with issues of class imbalance on huge datasets by adopting a novel approach to select samples for the creation of synthetic instances in place of the traditional approach used to find the k-nearest neighbor in SMOTE. Experiments they performed on some datasets indicate that the proposed strategy is successful in minimising the effects of class imbalance. Also, a Genetic Algorithm-based sampling technique was presented by [4], to identify minority instances that are suitable for resampling. The approach enabled the selection of best minority instances as the starting parents for resampling. According to their findings, dealing with extremely imbalanced datasets poses several challenges for class-imbalance learning using their model.

Algorithm-level techniques generally attempt to address the imbalance problem through the perturbation of learning paradigms or tuning of certain parameters of learning algorithms. These are mostly divided into ensemble and cost-sensitive techniques. Cost-sensitive techniques solve the problem of class imbalance by associating different costs to misclassifying instances in the different classes [8]. Higher cost is often associated with misclassification of the minority class instances. To improve prediction, ensemble approaches combine the strength of multiple models in order to obtain optimum results [20]. To establish the viability of some selected learning models when employed in ensemble learning, a

Boosting-based Ensemble Framework (BEF) was presented and evaluated by [21]. They considered ten (10) base classifiers across all ensemble learning techniques and reported that this approach led to improved classification results.

Traditional Hybrid approaches mostly leverage a combination of data-level techniques and algorithmic level techniques, making them distinct and effective solutions for addressing the challenges posed by imbalanced datasets [22]. However, recent trends to hybridisation have seen the combination of different data-level approaches to form hybrid sampling methods. A typical example in this regard is the work presented in [23], where a SMOTE-Nominal and Continuous (SMOTE-NC) approach is combined with Random Undersampling (RUS) to from a hybrid sampling method. They compared the performance of their approach with irregular oversampling, arbitrary undersampling, and RUS as base sampling strategies. The evaluation was done using a single classifier, the Random Forest, which recorded improved performance with the hybrid approach. Similarly, a Clustering and Distance-Based Imbalance Learning Model (CDEILM) and a Cluster Size-Based Resampling (CSBR) techniques were presented by [24]. CDEILM combines distance-based undersampling, feature selection and ensemble learning, while CSBR preserves the original distribution of the majority class. The experiments were conducted on two sets of publicly available datasets and the outcome showed that both CDEILM and CSBR approaches outperformed the benchmark methods.

While these techniques have been introduced to enhance the performance of classifiers on imbalance datasets, the need for addressing the class imbalance problem while considering the internal distribution of instances within classes which has a potential of influencing classifier performance particularly on extremely imbalance datasets has not been extensively explored. The goal of this study is therefore, to propose, implement and evaluate a dual sampling technique which has the potential of maintaining the internal distribution of instances within classes in order to minimise information loss due to excessive undersampling and the creation of redundant data in the case of oversampling.

2 Methodology

The methods employed in this study as well as the proposed Dual Sampling Technique (DST) is outlined in this section. These include a summary of the datasets, data sampling techniques, classification algorithms, and performance metrics used to evaluate the performance of the classification models.

2.1 Datasets

Five (5) benchmarked datasets sourced from the Kaggle repository [25], are used are used for all experiments reported in this study. These datasets are chosen because they have been widely used in similar studies and present varying degrees of imbalance. Attempts are made to ensure the selected datasets are diverse and representative enough to confirm the ability of the proposed technique to solved the imbalance problem in varied fields. A summary of the datasets showing the distribution of majority and minority class instances is presented in Table 1.

Table 1: Summary of Datasets

Dataset	Attributes	Instances	Positive Instances		Negative Instances	
			Number	%	Number	%
Credit Card	31	284,807	492	0.17	284,315	99.83
Diabetes	10	100,000	8,500	8.50	91,500	91.50
Telecom Churn	20	3,333	483	14.49	2,850	85.51
MetroPT3	16	1,516,948	95,406	6.29	1,421,542	93.71
Fraud Detection	12	6,362,620	8,213	0.13	6,354,407	99.87

2.2 Proposed Technique

The proposed technique dubbed Dual Sampling Technique (DST), employs a new sampling technique called the Average Oversampling Technique (AOT)and random undersampling to form a dual sampling approach. Applying both oversampling and undersampling in the proposed technique is aimed at reducing the inherent disadvantages of both techniques when applied excessively alone in extremely imbalance datasets. Additionally, performing sampling in the proposed technique at the cluster level is an attempt to ensure that the original distribution of instances in the dataset is maintained.

Sampling in DST involves a few steps: (1) the elbow method in K-Means algorithm is used to determine the optimal number of clusters k, in a training dataset, which is then clustered into the k clusters using the K-Means algorithm, (2) the number of positive and negative instances in each cluster are determine and the appropriate sampling rates computed and sampling performed.

For each cluster, $i \in k$, if the number of majority instances, maj, and minority instances, min, are such that their ratio, $r = maj/min$, falls within 0.2 and 1, the cluster is returned without sampling. Otherwise, if r is less than 0.2, the majority class instances in that cluster are considered noisy and discarded while retaining only minority instances. Secondly, if r is such that $1 < r \leq 1.25$, that is, the majority instances are at most 25% more than the minority instances in a cluster, the average oversampling technique described below is used to oversample the minority class instances in that cluster by a sampling rate defined as $(r - 1) \times 100$. Sampling by this ratio increases the number of minority instances in the cluster to almost the same as the majority instances. This is ideal because the cluster is not considered extremely imbalanced and as such the increase in minority instances will not excessively increase the size of minority class instances, which can affect model training time and may also lead to over-fitting.

On the other hand, if the imbalance ratio in a cluster is more than 1.25, that is, the majority instances are more by at least 26%, then both random undersampling and the average oversampling techniques are applied simultaneously to this cluster. The undersampling rate in this regard is set to $0.35\,(r-1)$, which implies that about 35% of the excess majority instances in the given cluster are randomly discarded. With respect to the AOT, the oversampling rate is set to $0.45\,(maj-min)$, implying that the minority instances are increased by about 45% of the excess majority class instances. The aim in this regard is not to ensure class parity but to minimise the imbalance ratio. However, if a cluster contains only majority instances, the difference between the majority and minority instances in the entire training set is obtained and undersampling of the cluster instances is performed using a rate equivalent to 45% of this difference.

As shown in Algorithm 2, AOT creates new minority instances in clusters by computing averages. In a given cluster, if the centroid is a minority class instance, then minority instances equal to the oversampling rate are randomly selected and each is averaged with the centroid (sample head) to form new instances. The selection is done such that the cluster head and its exact replicas, if they exist, are not used to generate new instances. If the number of minority instances in a cluster is less than the oversampling rate, the entire instances are used to create new instances and the process repeated each time with the new instance inclusive until the desired number of minority instances are obtained. In the case where the centroid is not a minority instance, a random minority instance in the cluster is selected to serve as the cluster head and new samples created accordingly. A pseudocode of the proposed sampling technique is shown in Algorithm 1.

Algorithm 1: Dual Sampling Technique

```
Input: Get the training dataset
Apply the Elbow method to get optimal number of clusters $k$, in the training dataset
Cluster training dataset into $k$ clusters using the K-Means algorithm
for $i$ to $k$ : do
    Determine the number of minority (min) and majority (maj) instances in $i$
    Compute, $r = \frac{maj}{min}$
    if $r \leq 1$ then
        if $r \geq 0.20$ then
            return Cluster $i$
        else
            return only minority instances in $i$
        end if
    else
        if $r \leq 1.25$ then
            OverPer = $(r - 1) * 100$
            Oversample minority in $i$ using Alg. 2
        else
            OverPer = $0.35 * (maj - min)$
            Oversample minority in $i$ using Alg. 2
            UnderPer = $0.45 * (maj - min)$
            Randomly undersample majority instances in $i$ by UnderPer
        end if
    end if
end for
Return Balanced training data
```

Algorithm 2 : Average Oversampling Technique

```
Input: Data in cluster $i$
Input: Percentage of instances to oversample, $OverPer$
Input: Number of minority instances in $i$, $min$
Compute number of instances to sample from $i$: $m = OverPer * min$
Obtain from $i$, the $m$ closest instances, $clIns$ to the cluster head, $ch$
for $j$ = 1 to $m$ do
    $syntheticIns[j] = \frac{ch + clIns[j]}{2}$
    Append to $i$, syntheticIns[$j$]
end for
Return: Balanced cluster, $i$
```

2.3 Experimental Setup

2.3.1 Sampling Techniques

Four commonly used sampling techniques are used in this study as the foundation to measure the effectiveness of DST. These techniques include Random Undersampling (RUS), Random oversampling (ROS), Synthetic Minority Oversampling Technique (SMOTE) and Cluster-Based Undersampling (CBU), which are all available as libraries the in Jupyter Notebook.

2.3.2 Classification Algorithms

In this study, three classification algorithms, Random Forest (RF), Decision Tree (DT), and KNearest Neighbor (KNN), were employed. These algorithms are chosen because they have been widely used in literature for similar studies and therefore serve as a strong foundation for the assessment of DST. Their implementation in the sci-kit-learn library is used. Also, default parameters of the algorithms are used except where otherwise specified. The stratified-10-fold-cross validation is adopted for all training and evaluation processes. This stratified approach is adopted to ensure that approximate number of minority instances are in each fold to avoid the situation of having some folds without minority instances.

2.3.3 Performance Metrics

Evaluating the performance of a Machine Learning model is a crucial step in creating an effective model. Area Under ReceiverOperating Characteristics Curve (AUC) and Geometric Mean (G-Mean) are the main metrics used in this study mainly because of their suitability for measuring classifier performance on imbalance datasets [26, 27].

3 Results and Discussion

The experimental results are presented and discussed in this section. The performance of the classification models trained after using the DST is presented and compared to when no sampling is done prior to training labelled “NONE” and four (4) existing sampling techniques; two undersampling (RUS and CBU) and two (2) oversampling techniques (ROS and SMOTE).

3.1 Experimental Results

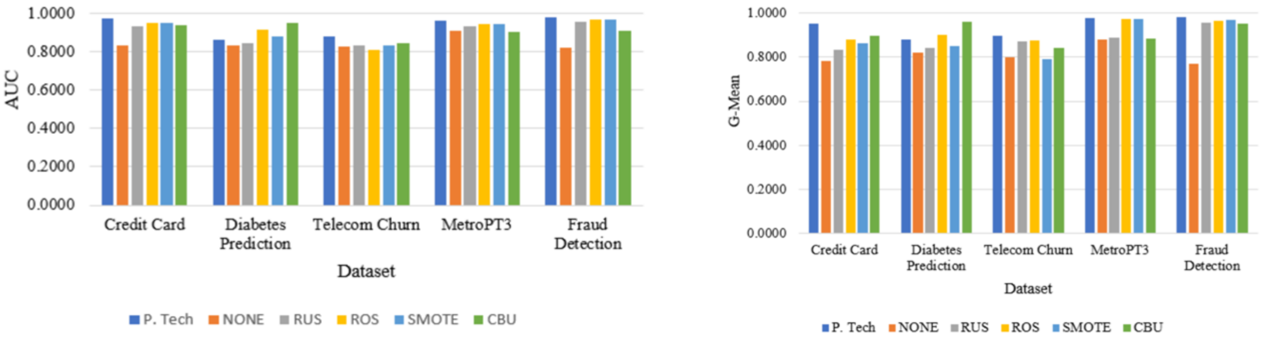
The experimental results are shown in Tables 2, 3 and 4 for Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbor (KNN) classifiers, respectively. Each table shows the AUC and G-Mean performance measures of the models in respect of the sampling techniques and datasets used. For each classification model and dataset, the performance measure of the sampling technique that yields the highest result is boldfaced.

Table 2: Experimental Results: Random Forest

AUC Measure						
Dataset	DST	NONE	RUS	ROS	SMOTE	CBU
Credit Card	**0.9732**	0.8321	0.9340	0.9521	0.9488	0.9401
Diabetes	0.8614	0.8343	0.8437	0.9134	0.8782	**0.9508**
Telecom Churn	**0.8797**	0.8290	0.8328	0.8091	0.8347	0.8431
MetroPT3	**0.9601**	0.9097	0.9319	0.9465	0.9457	0.9059
Fraud Detection	**0.9819**	0.8239	0.9563	0.9662	0.9675	0.9068
G-Mean Measure						
Credit Card	**0.9527**	0.7827	0.8319	0.8788	0.8630	0.8969
Diabetes	0.8797	0.8198	0.8410	0.9008	0.8490	**0.9608**
Telecom Churn	**0.8982**	0.7972	0.8714	0.8752	0.7925	0.8421
MetroPT3	**0.9764**	0.8807	0.8864	0.9728	0.9717	0.8859
Fraud Detection	**0.9819**	0.7685	0.9561	0.9662	0.9675	0.9505

Table 2 and as illustrated in Figure 1 (a) and (b), respectively, are AUC and G-Mean performance measures of the RF model given the various sampling techniques and datasets. As shown in the table, the proposed technique, DST, yielded the highest performance in four (4) out of the five (5) datasets in both AUC and G-Mean. AUC values of 0.9732, 0.8797, 0.9601 and 0.9819 while G-Mean values of 0.9527, 0.8982, 0.9764 and 0.9819 are obtained on the Credit Card, Telecom Churn, MetroPT3 and Fraud

Detection datasets respectively. However, CBU recorded the highest AUC value of 0.9508 on Diabetes Prediction dataset while DST performed marginally better than NONE and RUS. Similarly, using the G-Mean, DSToutperformed all other technique in four (4) out of the five datasets.



(a) (b)

Figure 1: Performance of Random Forest: (a) AUC and (b) G-Mean.

The experimental results for the Decision Tree algorithm are shown in Table 3 and illustrated graphically in Figure 2. These include the AUC and G-Mean measures of DST, NONE and the other four sampling techniques. Considering the AUC metric, DST recorded the highest scores; 0.8729, 0.9532 and 0.9806 in the Diabetes, MetroPT3 and Fraud Detection datasets, respectively. However, ROS outperformed the other techniques with a score of 0.9599 on the Credit Card dataset while CBU had the highest score of 0.9150 on the Telecom Churn dataset. Also, using the G-Mean measure, DST recorded the highest score on all the five datasets; i.e. 0.9998 on the Credit Card dataset, 0.8740 on Diabetes Prediction, 0.9269 on Telecom Churn, 0.9482 on MetroPT3 and 0.9806 on the Fraud Detection dataset.

Table 3: Experimental Results: Decision Tree

AUC Measure						
Dataset	DST	NONE	RUS	ROS	SMOTE	CBU
Credit Card	0.8524	0.8066	0.8739	**0.9599**	0.9497	0.9409
Diabetes Prediction	**0.8729**	0.8436	0.8237	0.8564	0.8518	0.8396
Telecom Churn	0.8581	0.7957	0.7858	0.8546	0.7425	**0.9150**
MetroPT3	**0.9632**	0.9159	0.9316	0.9450	0.9450	0.9417
Fraud Detection	**0.9806**	0.8339	0.9458	0.9531	0.9486	0.9470
G-Mean Measure						
Credit Card	**0.9998**	0.7539	0.8635	0.9454	0.9296	0.9268
Diabetes Prediction	**0.8740**	0.8362	0.8215	0.8636	0.8239	0.8245
Telecom Churn	**0.9269**	0.7722	0.7838	0.8152	0.6970	0.9144
MetroPT3	**0.9482**	0.9017	0.9155	0.9307	0.9307	0.9045
Fraud Detection	**0.9806**	0.7823	0.9455	0.9529	0.9486	0.9467

Table 4: Experimental Results: K-Nearest Neighbor

AUC Measure						
Dataset	DST	NONE	RUS	ROS	SMOTE	CBU
Credit Card	0.4142	0.2130	0.4057	**0.5198**	0.4956	0.3093
Diabetes Prediction	**0.7554**	0.4770	0.3455	0.5017	0.4887	0.3257
Telecom Churn	**0.6203**	0.4144	0.3347	0.5411	0.5481	0.3606
MetroPT3	**0.9589**	0.7386	0.8867	0.8414	0.8413	0.4337
Fraud Detection	0.9017	0.7698	0.8863	0.8486	**0.9208**	0.8486
G-Mean Measure						
Credit Card	0.3750	0.1929	0.2503	0.1936	0.1931	**0.3820**
Diabetes Prediction	**0.7159**	0.3466	0.5529	0.6625	0.5594	0.5471
Telecom Churn	**0.4950**	0.2579	0.2100	0.3371	0.3285	0.1820
MetroPT3	**0.9543**	0.6319	0.8332	0.7901	0.7901	0.2916
Fraud Detection	0.8003	0.7349	0.8845	0.8482	**0.9205**	0.3421

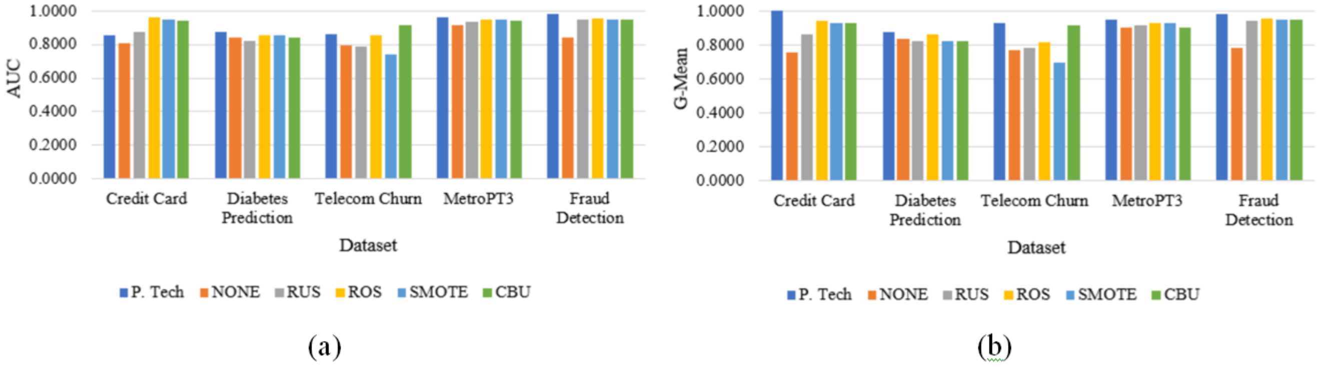


Figure 2: Performance of Decision Tree: (a) AUC and (b) G-Mean.

The experimental results of the K-Nearest Neighbor model obtained on the various datasets using the data sampling techniques are shown in Table 4 and graphically illustrated in Figure 3. From Table 4, the proposed technique, DST yielded the highest AUC values in three (3) out of five datasets considered. It yielded AUC scores of 0.7554, 0.6203 and 0.9589, respectively, on the Diabetes Prediction, Telecom Churn and MetroPT3 datasets. However, the DST recorded relatively lower AUC values on the Credit Card and Fraud Detection datasets, 0.4142 and 0.9017, respectively as against 0.5198 for ROS and 0.4956 for SMOTE on the Credit Card dataset and 0.9208 for SMOTE on the Fraud Detection dataset. Considering the G-Mean measure, DST again yielded the highest score of 0.7159 on the Diabetes Prediction, 0.4950 on Telecom Churn and 0.9543 on the MetroPT3 datasets while trailing behind SMOTE and CBU with scores of 0.9205 and 0.3820 for Fraud Detection and Credit Card datasets, respectively.

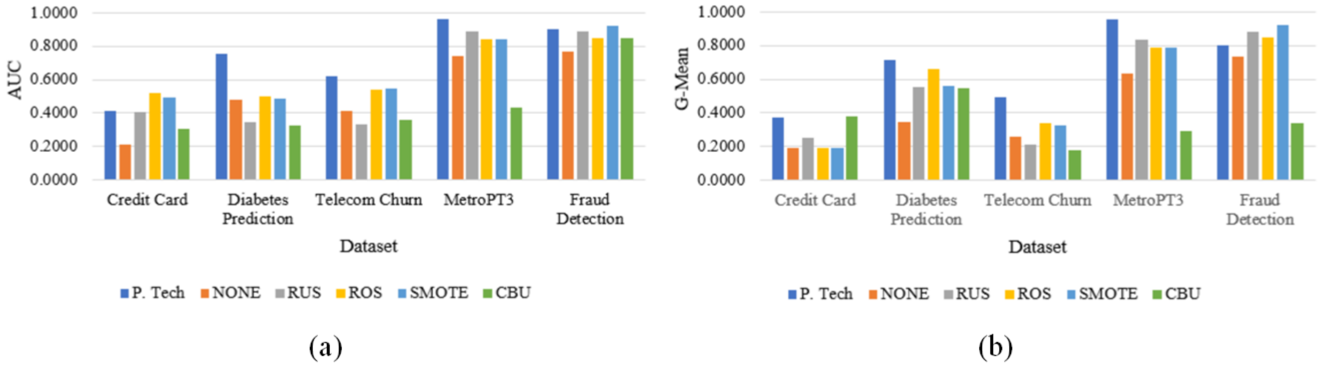


Figure 3: Performance of K-Nearest Neighbor: (a) AUC and (b) G-Mean.

3.2 Discussion

From the results presented above, the assertion that the proposed technique (DST) has the potential of improving classifier performance on imbalance datasets is confirmed. This is indicative from the fact that the use of DST resulted to improved classifier performance in all test cases considered when compared to the performance of classifiers trained without sampling the data. To illustrate numerically the gains made by DST against NONE, from Table 2, when the RF is used as the base classifier; DST resulted in minimum improvements of about 3.26% in AUC and 7.31% in G-Mean on the Diabetes prediction datasets and maximum improvements of about 19.18% in AUC and 27.77% in G-Mean on the Fraud Detection datasets, respectively. Similarly, from Table 3, where the DT is used as the base classifier, 3.47% and 4.52% minimum improvements in terms of AUC and G-Mean were recorded on the Diabetes Prediction dataset while the highest improvements of about 17.04% and 32.62% in terms of AUC and G-Mean were realised on the Fraud Detection and Credit Card datasets, respectively. Using the KNN as the base classifier as in Table 4, the least improvements of 17.07% and 8.90%, respectively, in terms of AUC and G-Mean were recorded on the Fraud Detection dataset. The KNN model obtained 58.36% AUC and 106.55% G-Mean performance improvements on the Diabetes Prediction dataset.

These improvements across the datasets using the various models suggest that, DST indeed has the potential to improve classifier performance on datasets with varying characteristics and imbalance ratios. This assertion is drawn based on the fact that there is no established trend in classifier performance given the dimensionality, size and most importantly, the imbalance ratios of the datasets with respect to the proposed technique. This characteristic of the DST suggests that it is suitable for all forms of imbalance learning tasks. What is rather evident is the resilience of the individual models to the imbalance problem as the RF and DT models tends to produce better performance without data sampling as against the KNN model, which produced relatively poor results both with and without sampling.

On the other hand, while DST resulted in improved classifier performance across all datasets considered, the four existing sampling techniques, RUS, ROS, SMOTE and CBU could not improve the classifiers performance on some datasets. For instance, as shown in Table 3, using ROS and CBU with RF classifier, resulted in marginally lower performance of about 2.46% and 0.42% AUC measure on the Telecom Churn and MetroPT3 datasets. And also, 0.59% lower G-Mean measure on Telecom Churn when SMOTE is used prior to training as against the classifier's performance without sampling. Using the DT as the base classifier, RUS resulted in lower AUC measures of about 2.42% and 1.26% on the Diabetes and Telecom Churn datasets, respectively. Similarly, SMOTE and CBU resulted in AUC measures of about 7.16% and 0.48% lower on the Telecom Churn and Diabetes datasets, when compared to NONE. The G-Mean measures of the DT models obtained using the existing techniques and NONE followed a similar trend, as RUS, SMOTE and CBU resulted to about 1.79%, 1.49%, and 1.42% lower G-Mean measures when compared to NONE on the Diabetes dataset. Considering the KNN models, RUS and CBU resulted to lower performance measures when compared to NONE. From Table 4, RUS resulted to 38.06% and 23.81% lower AUC values on the Diabetes and Telecom Churn datasets and 22.81% lower G-Mean on the Telecom

Churn dataset. CBU, on the other hand yielded 46.45%, 14.92%, and 70.30% lower AUC values on the Diabetes, Telecom Churn and MetroPT3 datasets, and 41.70%, 116.70% and 114.82% lower G-Mean values on the Telecom Churn, MetroPT3, and Fraud Detection datasets, respectively.

In a summary, the above establishes the potential of the proposed technique in solving the class imbalance problem since it has improved classifier performance across all the datasets considered when compared to NONE.

4 Conclusion

As part of efforts to address the class imbalance problem in machine learning, a Dual Sampling Technique that combines a novel average oversampling technique with random undersampling is proposed, implemented and assessed in this paper. The proposed technique effectively addresses the imbalance problem by taking into account the local proximity of both minority and majority class instances as part of its process while intelligently maintaining the internal distribution of instances within classes as way of minimising the effects of within class imbalance. The performance of the proposed technique is compared to existing sampling techniques when RF, DT, and KNN are used as the baseline learning algorithms on five (5) benchmarked datasets. The results revealed that the proposed technique consistently enhanced the performance of the learning algorithms as measured using the AUC, and G-Mean metrics when no sampling is done prior to training and when four (4) existing sampling techniques (RUS, ROS, SMOTE, and CBU) are applied to the training datasets prior to training.

However, it is observed that the proposed technique and CBU become relatively computationally expensive when used on larger datasets with higher number of clusters. To confirm the efficiency and practicability and to unearth potential limitations of the proposed technique, it is recommended that further investigations be conducted using diverse datasets of varying sizes and levels of imbalance as well as different classification algorithms. Also, explorations geared towards minimising the computation requirements of the proposed technique resulting from the use of K-Means algorithm for clustering is highly recommended.

References

[1] Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., & Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110, 104743. https://doi.org/10.1016/j.engappai.2022.104743

[2] Devi, D., Namasudra, S., & Kadry, S. (2021). A boosting-aided adaptive cluster-based undersampling approach for treatment of class imbalance problem. *International Journal of Data Warehousing and Mining*, 16(3). https://doi.org/10.4018/IJDWM.2020070104

[3] García-Gil, D., Luque-Sánchez, F., Luengo, J., García, S., & Herrera, F. (2019). From big to smart data: Iterative ensemble filter for noise filtering in big data classification. *International Journal of Intelligent Systems*, 34(12), 3260–3274. https://doi.org/10.1002/int.22193

[4] Venkateswarlu, B., Poornima, K., Vasavi, R., & Vaishnavi, J. V. (2022). A study on class imbalance problem using genetic algorithm. In *Proceedings of the 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1709–1714). https://doi.org/10.1109/ICSSIT53264.2022.9716371

[5] Saleh, M., Shahabadi, E., Tabrizchi, H., & Kuchaki, M. (2021). A combination of clustering-based under-sampling with ensemble methods for solving imbalanced class problem in intelligent systems. *Technological Forecasting & Social Change*, 169, 120796. https://doi.org/10.1016/j.techfore.2021.120796

[6] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., Hawalah, A., & Hussain, A. (2016). Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *IEEE Access*, 4, 7940–7957. https://doi.org/10.1109/ACCESS.2016.2619719

[7] Weiss, G. M. (2013). Foundations of imbalanced learning. In *Imbalanced Learning: Foundations, Algorithms, and Applications* (pp. 13–41). Wiley. https://doi.org/10.1002/9781118646106.ch2

[8] Chen, W., Yang, K., Yu, Z., Shi, Y., & Chen, C. L. P. (2024). A survey on imbalanced learning: Latest research, applications and future directions. *Artificial Intelligence Review*, 57, 137. https://doi.org/10.1007/s10462-024-10759-6

[9] Lin, C., Tsai, C.-F., & Lin, W.-C. (2023). Towards hybrid over- and under-sampling combination methods for class imbalanced datasets: An experimental study. *Artificial Intelligence Review*, 56(2), 845–863. https://doi.org/10.1007/s10462-022-10186-5

[10] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. https://doi.org/10.1613/jair.953

[11] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.

[12] Chen, Z., Lin, T., Xia, X., Xu, H., & Ding, S. (2018). A synthetic neighborhood generation-based ensemble learning for the imbalanced data classification. *Applied Intelligence*, 48, 2441–2457. https://doi.org/10.1007/s10489-017-1088-8

[13] Datta, S., & Arputharaj, A. (2018). An analysis of several machine learning algorithms for imbalanced classes. In *Proceedings of the 5th International Conference on Soft Computing & Machine Intelligence* (pp. 22–27). https://doi.org/10.1109/ISCMI.2018.8703244

[14] Sowah, R. A., Kuditchar, B., Mills, G. A., Acakpovi, A., Twum, R. A., Buah, G., & Agboyi, R. (2021). HCBST: An efficient hybrid sampling technique for class imbalance problems. *ACM Transactions on Knowledge Discovery from Data*, 16(3), 1–37. https://doi.org/10.1145/3488280

[15] Tsai, C.-F., Lin, W.-C., Hu, Y.-H., & Yao, G.-T. (2019). Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences*, 477, 47–54.

[16] Lin, W.-C., Tsai, C.-F., Hu, Y.-H., & Jhang, J.-S. (2017). Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409, 17–26. https://doi.org/10.1016/j.ins.2017.05.008

[17] Kang, Q., Shi, L., Zhou, M., Wang, X., Wu, Q., & Wei, Z. (2017). A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2017.2755595

[18] Nugraha, W., Maulana, M. S., & Sasongko, A. (2020). Clustering-based undersampling for handling class imbalance in C4.5 classification algorithm. *Journal of Physics: Conference Series*, 1641, 012014. https://doi.org/10.1088/1742-6596/1641/1/012014

[19] Rodríguez-Torres, F., Martínez-Trinidad, J. F., & Carrasco-Ochoa, J. A. (2022). An oversampling method for class imbalance problems on large datasets. *Applied Sciences*, 12(7), 3424. https://doi.org/10.3390/app12073424

[20] Hamad, R. A., Kimura, M., & Lundström, J. (2020). Efficacy of imbalanced data handling methods on deep learning for smart homes environments. *SN Computer Science*, 1(4), 204. https://doi.org/10.1007/s42979-020-00211-1

[21] Qian, M., & Li, Y.-F. (2022). A weakly supervised learning-based oversampling framework for class-imbalanced fault diagnosis. *IEEE Transactions on Reliability*, 71(1), 429–442. https://doi.org/10.1109/TR.2021.3138448

[22] Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 1–54. https://doi.org/10.1186/s40537-019-0192-5

[23] Wongvorachan, T., He, S., & Bulut, O. (2023). A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. *Information*, 14(1). https://doi.org/10.3390/info14010054

[24] Kou, G., Chen, H., & Hefni, M. A. (2022). Improved hybrid resampling and ensemble model for imbalance learning and credit evaluation. *Journal of Management Science and Engineering*, 7(4), 511–529. https://doi.org/10.1016/j.jmse.2022.06.002

[25] Kaggle. (2009). *Kaggle: Your home for data science*. https://www.kaggle.com

[26] Acuña, E., & Rodríguez, C. (2005). An empirical study of the effect of outliers on the misclassification error rate. Manuscript submitted for publication.

[27] Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 179–186). Morgan Kaufmann.