A Comparative Study for Different Resampling Techniques for Imbalanced datasets

Alaa M. Elsobky, Arabi El. Keshk, Mohammed G. Malhat
Department of Computer Science
Faculty of computers and information
Menofia university
Menofia, Egypt
alaa.elsobky@ci.menofia.edu.eg, arabi77Staff@ci.menofia.edu.eg, mohamed.gaber@ci.menofia.edu.eg

Abstract—The imbalanced data is a significant challenge for researchers in supervised machine learning. Current data mining algorithms are not effective for processing imbalanced data. In fact, this problem reduces classification accuracy because the prediction of minority classes is inaccurate. The classification of imbalanced data is the major challenge that has received significant attention. Therefore, The use of sampling techniques to improve classification performance has been a significant consideration in related work. In this paper, a comparative study of six different sampling algorithms is performed. The employed sampling algorithms are from different sampling techniques: two oversampling algorithms, two under sampling algorithms, and two combination algorithms between oversampling and under sampling. The techniques used in oversampling are random oversampling and SMOTE, while under sampling techniques are random under sampling and a near miss. A combination of oversampling and under sampling techniques is SMOTE TOMEK and SMOTEEN. This comparative study aims to examine the impact of the employed sampling method. Algorithms on the performance of three classifiers: SVM, KNN, and logistic regression. Cross-validation experiments on 8 standard datasets show that the SMOTEEN sampling The algorithm achieves significant improvements compared with other typical algorithms.

Keywords— Imbalanced data, resampling techniques, SMOTE, SMOTEEN, SMOTE Tomek, Nearmiss

I. Introduction

Data is one of the fastest-growing sectors globally. Scientists need to collect and analyze huge quantities of data to generate actionable insights that an organization can use to reinforce its different aspects. It is a broad concept with a lot of advantages. To understand big data, you must become familiar with its root characteristics. Understanding the characteristics of big data [1] is vital to understanding how it works and how you can use it. Volume refers to the amount of data that you have, which rises substantially in a short time. Velocity is the speed of data processing. High velocity is critical for the performance of any big data process. Value is the benefit that your organization gets from the data. Variety refers to the different types of big data (Structured, Unstructured, and semi-structured), which affect performance. You need to organize that data to manage its variety properly. Veracity refers to the accuracy of your data.

Veracity is among the most important big data characteristics, as low veracity [2] can greatly damage the accuracy of your results, for example, if the data is imbalanced. Through big data [3], the mitigation of class imbalance causes a greater challenge because of the difference and complex structure of the relatively much larger datasets [4]. An imbalanced classification [5] problem is an example of a classification problem where the distribution of examples through the recognized classes is unfair or skewed. The distribution can differ from a slight bias to an acute imbalance [6].

Imbalanced classifications [7] are a challenge for predictive modeling, as most of the machine learning algorithms used for classification were designed around the hypothesis of a similar number of examples for each class [8]. This assumption is made in models that have poor predictive performance, for the minority class [9]. This is a problem because in most cases, the minority class is more significant, and therefore the problem is more critical to classification errors for the minority class than the majority class [10]. Therefore, it is necessary to deal with this data imbalance problem when training machine learning algorithms [11]. Machine learning is used to keep up with the ever-growing and ever-changing stream of data and present continuously evolving and valuable insights [12].

Most researchers are interested in imbalanced data, as Sara et al. [13] used 15 cancer imbalanced data sets, 18 over- and under-sampling techniques, and four different classifiers. The SMOTE has achieved the best accuracy results. For detecting accidents, Parsa et al. [14] apply SMOTE as oversampling, support vector machines, and Probabilistic Neural networks. It achieves the best result by using an AUC of 90%. And what is worth mentioning is that most of them don't use a combination of sampling techniques. Multiclass wasn't taken into consideration. This paper tries to study the impact of the imbalanced data problem on the machine learning models' performance. It uses different resampling methods and machine learning classifiers to solve the imbalanced data problem and compares these methods. The vital processes of this research as compared to similar research works include: Implementing and comparing different resampling methods, namely random oversampling, SMOTE, random under sampling, near miss, SMOTE TOMEK, and SMOTEEN Applying the model validation method, which is a K-fold cross-validation method, to perform the validation. Using different machine learning models such as Support Vector Machine (SVM), logistic regression, and K-Nearest Neighbor to compare the performance of resampling methods Using various evaluation measure methods such as Recall, Precision, and F1-Score to measure the performance of the implemented models Showing the impact of the resampling methods on the classifier's performance. Analyzing the differences between resampling methods and determining the best method among others. The remainder of the paper is organized as follows: In Section 2, an overview of previous related research is described. Section 3 provides other existing sampling methods, including

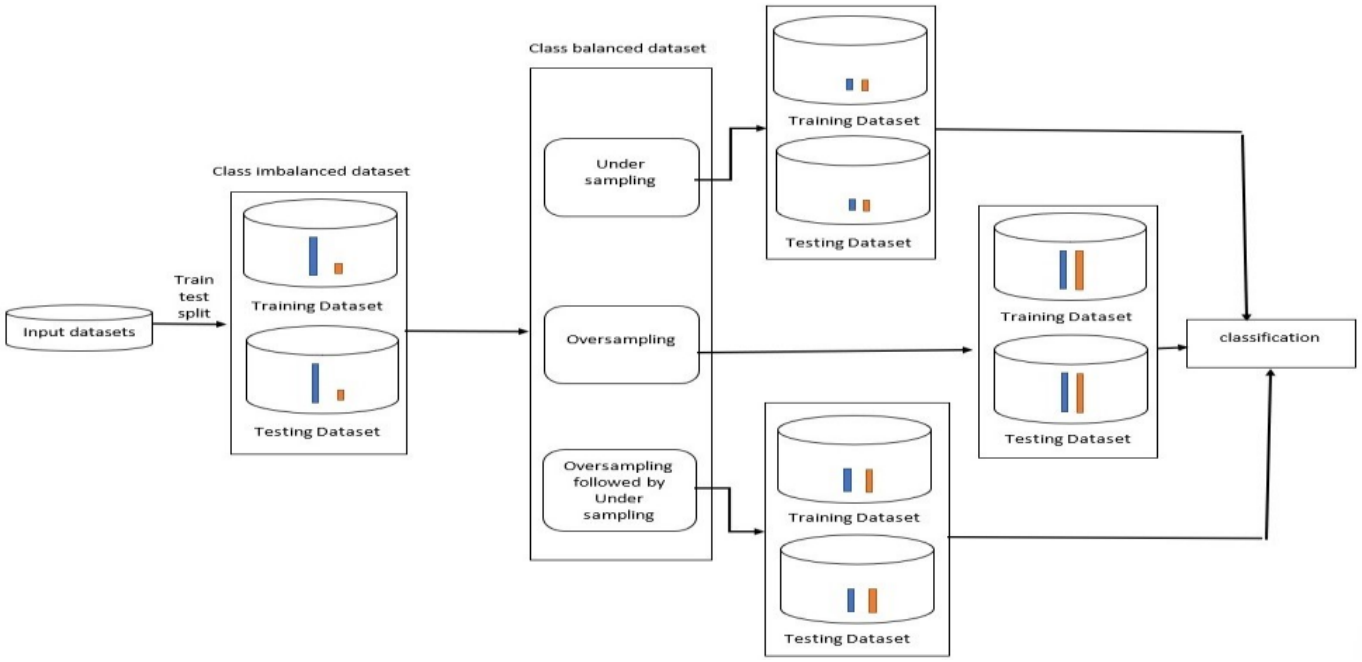

Fig 1 . Block diagram to illustrate the class imbalance problems

oversampling, under sampling, and merging between them. The comparison method is described in detail and evaluation matrices are provided in Section 4, while some conclusions and future work are drawn in Section 5.

II. Related work

The problem of learning from imbalanced data concerns many researchers. The solution is classified in two ways: data level and algorithm level. [15], [16]. At the data level approach, the training data is modified by addition or elimination to balance the dataset, called resampling, as shown in Fig. 1. Resampling techniques have three approaches. First, oversampling means adding more observations for the minority class. Second, under sampling eliminates instances from the majority class. The third is a combination of oversampling and under sampling [17]. At the algorithm level, [18] is creating a new algorithm or modifying an existing one to deal with imbalanced data without any modification.

The problem of data imbalance is widespread in many applications and fields. At the medical level [13], [19]–[22], the problem of diagnosing rare diseases, for example, cancer, is much smaller than the number of healthy people. Sara et al. [13] used oversampling and under sampling to check the impact of balancers on the performance of classifiers. The classifiers are named RIPPER [23] , multi-layer perceptron (MLP) [24] , k-nearest neighbors (KNN) [25], and C4.5 [26] decision tree classifiers. They used datasets from the SEER program for specific types of cancer. The results of the study summarize the effect of classifiers on a group of pre-processing techniques to determine the best balancer and classifier for specific datasets. The safe-level SMOTE [27] achieves the best accuracy results of 0.802 for oversampling methods, and the random under sampling (RUS) [28] achieves the best accuracy results of 0.791 for under sampling methods.

In the field of education [29]–[31], Ghorbani [29] et al used a set of oversampling techniques such as synthetic minority oversampling technique(SMOTE) [32], Borderline SMOTE [33], random oversampling [34], SMOTE and edited nearest neighbor(SMOTE-ENN) [35], SVM-SMOTE, and SMOTE-Tomek to study their effect on imbalanced data with different classifiers as Random Forest [36], K-Nearest-Neighbor [37], Artificial Neural Network [38], XG-boost [39], Support Vector Machine [40] (Radial Basis Function), Decision Tree [41], Logistic Regression [42], and Naïve Bayes [43]. This study resulted in using SVM-SMOTE [44] as a resampling method and the Random Forest model as a classification technique to achieve the best result according to predicting students' performance.

In addition, some researchers [14], [45]–[47] turn to the topic of accident analysis so that they can discover the accident and deal with it, rescue the injured as soon as possible, and not disturb road users. Parsa et al. [14] used a dataset containing 85214 instances: 85,182 non-accident cases and 32 accident cases. The dataset is resampled by the SMOTE [32], oversampling technique. They used a support vector machine and a probabilistic neural network (PNN) [48] is used for pattern recognition problems and classification. The model is trained to detect an accident after five minutes, which it achieves at 90% using PNN.

Overall, most of the research previously focused on finding the appropriate sampling for a specific field. It relied on the implementation of bi-classes and neglected multi-classes. It also did not benefit from the idea of merging, as there was oversampling for the minority class and under sampling for the majority class at the same time. Most of the authors don't consider any modifications to classification algorithms.

III. Proposed Work

Machine learning techniques often fail or give misleading performances on classification datasets with an imbalanced class division. That's because many machine learning algorithms are designed to work on classification data with a similar number of observations for each class. When this is not the case, algorithms can pick up that very few

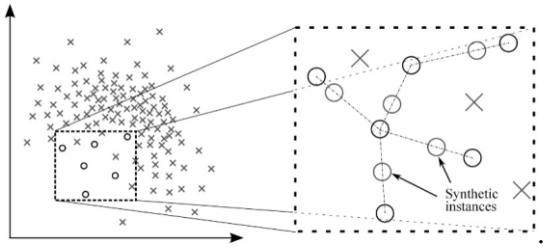

Fig 2 . Example of using SMOTE to add new sample between two existence samples

examples are not important and can be neglected in order to achieve perfect performance. Sampling techniques are a solution to the imbalanced learning problem; they simply modify imbalanced data so it will have a balanced distribution. Sampling techniques commonly used in imbalance learning are oversampling, under sampling, or a combination of them.

A. oversampling technique

1) Random OverSampling: The algorithm 1 describes [49] the random number of observations in the minority class and then copies them randomly to balance the dataset. The duplicate instances were put randomly in the dataset.

Algorithm 1 Random OverSampling algorithm

Input:Imbalanced data M, Number of extra observations Z.

Output:Modified balanced data S

```
procedure Random OverSampling algorithm
    while Z = 0 do
        Select random one of the minority data M, call this x'
        append x' to S.
        Z = Z − 1.
    end while
end procedure
```

2) SMOTE: The synthetic Minority Oversampling Technique [50] is improving duplicate instances randomly by creating synthetic ones. Fig. 2. gets new instances between existing minority instances. It creates synthetic observations along the line joining them to their k nearest neighbors, as shown in the algorithm 2.

B. under sampling technique

1) Random under sampling: The algorithm 3 eliminates the random number of observations in the majority class. This process was repeated numerous times to reach a suitable number of instances until classes were balanced [51].

2) Near miss: The Algorithm 4 calculates the distance between each point in the majority class and the minority class, then chooses the shortest distance in the minority class, which will be removed from the dataset [52].

C. Combination of oversampling and under sampling technique

1) SMOTE Tomek: This technique 5 is a hybrid one based on the development of SMOTE by Tomek, as it cleans the data by eliminating boundary points in regions shown in Fig. 3. where it is unclear which of two or more classes.

Algorithm 2 SMOTE algorithm

Input:Imbalanced data M, Number of extra observations Z.

Output:Modified balanced data S

```
procedure SMOTE
    for i = 1, 2, ..., T do
        Find the k nearest (minority class M) neighbors of x_i
        while Z = 0 do
            Select one of the k nearest neighbors, call this x'
            Select a random number ℵ ∈[0,1].
            x" = x_i + ℵ(x' − x_i).
            append x" to S.
            Z = Z − 1.
        end while
    end for
end procedure
```

Algorithm 3 Random under sampling algorithm

Input:Imbalanced data M, Number of extra observations Z.

Output:Modified balanced data S

```
procedure Random under sampling algorithm
    while Z = 0 do
        Select random one of the majority data M, call this x'
        remove x' from S.
        Z = Z − 1.
    end while
end procedure
```

Algorithm 4 Near miss algorithm

Input:Imbalanced data (Minority Data M_i, Majority Data N_j), Number of extra observations Z.

Output:Modified balanced data S

```
procedure Near miss algorithm
    while Z = 0 do
        for i ← 1 to M do
            for j ← 1 to N do
                Calculate distance between N, M.
            end for
        end for
        Choose the shortest distance with M .
        The shortest distance refer to n class stored for elimination.
        Z = Z − 1.
    end while
end procedure
```

Algorithm 5 SMOTE TOMEK algorithm

Input:Imbalanced data (Minority Data M_i, Majority Data N_j), Number of extra observations Z.

Output:Modified balanced data S

```
procedure SMOTE TOMEK
    for i = 1, 2, ..., F do
        Find the k nearest (minority class) neighbors of x_i
        while Z = 0 do
            Select one of the k nearest neighbors, call this x'
            Select a random number ℵ ∈[0,1].
            x" = x_i + ℵ(x' − x_i).
            append x" to S.
            Z = Z − 1.
        end while
    end for
    T' = [T/100]
    while T' = 0 do
        Choose random data d from J
        if d then is nearest from M
            remove this point
        end if
        T' = T' − 1.
    end while
end procedure
```

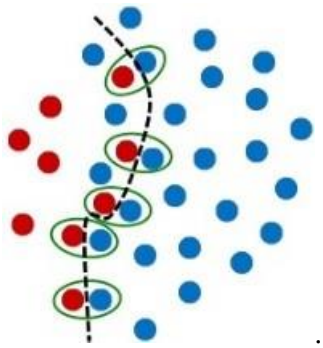

Fig 3 . A Borderline using tomek to remove the nearest sample of minority class

2) SMOTEEN: It is merging between oversampling and under sampling techniques by adding new observations for the minority class using SMOTE and removing existing observations from the majority class using Edited Nearest Neighbor (ENN). It depends on the idea of KNN. If there is a new observation in the dataset, it must be known to which class it belongs. We calculate each observation distance in KNN. For example, if there are two classes, black and white, as shown in Fig. 4., suppose a new observation is added to determine which class it will belong to. By using the KNN to calculate the distance between an observation and two classes, if the majority of the new observation's KNN belongs to the black class, then the observation will belong to the black class. It takes advantage of oversampling using SMOTE and filtering noise using ENN as under sampling techniques. In the beginning, algorithm 6 describes SMOTE. It chooses random data from the minority class. It calculates the distance between it and its k-neighbors. A random number between 0 and 1 is chosen and multiplied by the distance. The result will be added to the minority class as synthetic simple. Then, it describes ENN as KNN, which will be determined. It calculates the KNN of the observation between the others, then returns the majority class from the KNN. If they are different, they will be removed from the dataset.

Algorithm 6 SMOTEEN algorithm

Input:Imbalanced data (Minority Data M_i, Majority Data N_j), Number of extra observations Z.

Output:Modified balanced data S

```
procedure SMOTEEN
    for i = 1, 2, ..., F do
        Find the k nearest (minority class) neighbors of x_i
        while Z = 0 do
            Select one of the k nearest neighbors, call this x'
            Select a random number ℵ ∈[0,1].
            x" = x_i + ℵ(x' − x_i).
            append x" to S.
            Z = Z − 1.
        end while
    end for
    for each instance (i) do
        if the class of instance (i) != the majority class of k neighbors then Remove the Instance
        end if
    end for
end procedure
```

IV. Experimental results

This section illustrates computer specifications, the package used, and imbalanced datasets. Different evaluation matrices are used to help efficiently evaluate techniques and implement them. Experimental results show the effect of using various sampling techniques and machine learning classifiers.

A. Setup

This research has used a processor, an Intel Core i7, operating system, Microsoft Windows 11 Professional x64, and 16 GB of RAM. The software is designed as a Python 3.9 package, basically built on the machine learning algorithms of the sklearn , imblearn package for oversampling and under sampling.

B. datasets

The datasets include a set of features for real-world applications and are accessible from the KEEL dataset [53], which was last modified in February 2010. We apply algorithms of sampling to a dataset with binary and multiclass features ranging from 4 to 21, and finally, an imbalanced ratio (IR) ranging from 1.5 to 129.43. Table 1 shows the distribution ratio of pre-classified classes as the imbalanced ratio of the

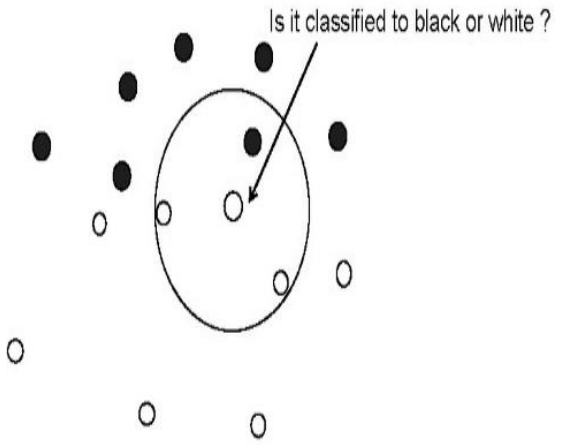

Fig 4 . Using SMOTEEN to identify sample that can be remove depend on its neighbor

Table 1 . Summary of datasets used in the experiments.

Dataset	Features	Minority class/ Majority class	Imbalance ratio
Wine	13	48/71	1.5
Vehicle1	18	212/417	2.9
Ecoli2	7	52/284	5.46
Balance	4	49/288	5.8
Ecoli3	7	35/301	8.6
Glass2	9	17/197	11.59
Thyroid	21	17/666	36.94
Abalone1	9	32/4142	129.43

majority and minority classes, the number of features, and the number of instances for each dataset.

This paper shows the effect of using various resampling methods and classifications on imbalanced data. Additionally, it determines the best classifier with the best resampling model compared to the others The raw data used in this paper includes 8 different datasets from the Keel, as it is imbalanced data. Table 1. shows more detailed information about the 8 datasets.These datasets divide into two partitions: bi-classes that are positive and negative, or zero and one. Multi-class contains more than one category. In this paper, multi-class contains three categories. The datasets are classified according to KNN [25] classifier, logistic regression [42], and SVM [40] classifications. The three classifications are applied to the datasets individually, along with six resampling techniques applied with k-fold cross-validation. It should be indicated that the classification was carried out on imbalanced data first and then on balanced data to notice the effect of applying resampling as a solution to the imbalanced data. All presented models have been coded in Python, which is a high-level, interpreted, and general-purpose programming language. There is a more meaningful performance for imbalanced data than for accuracies such as F1_score, Precision, and Recall.

C. Evaluation metrix

Performance techniques play an important role in classification model evaluation. Traditional evaluation is not acceptable for the imbalanced data; calculating accuracy according to the model generally without taking into consideration each class in the model makes the accuracy misleading. To quantitatively evaluate the classification performance in the imbalanced data, the confusion matrix is one of the performance techniques that helps us calculate the accuracy for each class individually. Table 2 shows a confusion matrix where the columns are the classified class and the rows are the actual class. In the confusion matrix [13], True Positives (TP) are the number of minority class instances correctly classified, True Negatives (TN) are the number of majority class instances correctly classified, False Positives (FP) are the number of negative instances incorrectly considered positive, and False Negatives (FN) are the number of positive instances incorrectly considered negative. Most of the time, the performance of the minority class is basic; precision, recall, and F1-score are measures for the minority class calculated by a confusion matrix. Precision is the ratio of the true positive to the number of actual positives according to equation 1 . Recall is the ratio of actual positive to predicted results according to equation 2 . Accuracy is the ratio of actual to total (actual and predicted) according to equation 3. Using accuracy in imbalanced data leads to misleading results because the data is distributed skewed, so we neglect it in this study. F1-score is the multiplication of Recall and Precision and the summation of them according to equation 4.

Table 2 . confussion matrix.

		Predict	
		positive	negative
Actual	positive	TP	FN
	negative	FP	TN

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$F1 = \frac{2 * precision * Recall}{precision + Recall} \quad (4)$$

D. Results

SVM classification is applied with the resampling method, and it's evaluated by precision metrics shown in Table 3. which resulted in recognizing true positives according to all predictive positives. It can't find all true positives, but the ones that are classified as positive are very likely to be correct. Near miss achieves the best for wine and balance with an imbalance ratio of 1.5, 5.8, and contains 3 classes as it improves from 0.83 without sampling to 0.98 according to the balance data. Random under sampling and SMOTEEN achieve 0.95 for the ecoli2 dataset with an imbalanced ratio of 5.46; it could recognize a true positive in ecoli2 according to all predictive positives. SMOTEEN achieves the best performance as it combines the strategy of downsizing the majority class, removing the misclassification that is considered noise, and generating new points to balance the datasets. Without resampling, it is not a precise model; it wrongly detects many positives that aren't actually positives and may find a lot of positives, but its selection method is noisy.

Table 3 . Precision, Recall, and F1-score with SVM Classifier

Dataset	IR	without sampling	under sampling		Oversampling		combination	
			Near miss	Random UnderSampler	Random OverSampler	SMOTE	SMOTE Tomek	SMOTE ENN
Precision								
Wine	1.5	0.94	0.97	0.95	0.96	0.96	0.95	0.95
Vehicle1	2.9	0.64	0.56	0.71	0.76	0.83	0.82	0.96
Ecoli2	5.46	0.91	0.87	0.95	0.89	0.92	0.92	0.95
Balance	5.8	0.83	0.98	0.91	0.92	0.88	0.87	0.97
Ecoli3	8.6	0.45	0.47	0.87	0.9	0.91	0.93	0.99
Glass2	11.59	0.46	0.58	0.57	0.81	0.8	0.79	0.88
Thyroid	36.94	0.31	0.54	0.3	0.79	0.85	0.85	0.89
Abalone19	129.43	0.5	0.8	0.75	0.82	0.85	0.85	0.88
Mean		(6) 0.67	(5)0.73	(4) 0.77	(3) 0.83	(2) 0.84	(2) 0.84	(1) 0.9
Recall								
Wine	1.5	0.95	0.96	0.95	0.97	0.96	0.95	0.95
Vehicle1	2.9	0.63	0.56	0.71	0.75	0.8	0.79	0.95
Ecoli2	5.46	0.74	0.86	0.91	0.89	0.92	0.92	0.94
Balance	5.8	0.92	0.97	0.9	0.92	0.89	0.87	0.97
Ecoli3	8.6	0.5	0.57	0.88	0.89	0.91	0.92	0.95
Glass2	11.59	0.5	0.55	0.69	0.69	0.66	0.69	0.76
Thyroid	36.94	0.33	0.49	0.36	0.74	0.78	0.77	0.81
Abalone19	129.43	0.5	0.66	0.71	0.81	0.85	0.84	0.87
Mean		(6) 0.67	(5) 0.73	(4) 0.77	(3) 0.83	(2) 0.84	(2) 0.84	(1) 0.9
F1-score								
Wine	1.5	0.94	0.96	0.95	0.96	0.96	0.95	0.95
Vehicle1	2.9	0.63	0.56	0.71	0.75	0.79	0.79	0.95
Ecoli2	5.46	0.79	0.86	0.92	0.89	0.92	0.92	0.94
Balance	5.8	0.85	0.97	0.9	0.92	0.88	0.87	0.97
Ecoli3	8.6	0.47	0.44	0.87	0.89	0.91	0.92	0.95
Glass2	11.59	0.48	0.49	0.52	0.66	0.62	0.66	0.75
Thyroid	36.94	0.32	0.46	0.28	0.71	0.75	0.75	0.81
Abalone19	129.43	0.5	0.61	0.69	0.81	0.85	0.84	0.88
Mean		0.66(6)	0.71(5)	0.75(4)	0.82(3)	0.83(2)	0.83(2)	0.9(1)

The recall metric is shown in Table 3. is a true positive rate that shows how well the model can predict all samples in the dataset; it shows how many related samples are chosen. The random oversampling achieves 0.97 for the wine dataset, with an imbalanced ratio of 1.5. Near miss and SMOTEEN achieve 0.97 for the balance data, with an imbalanced ratio of 5.8 with multiclass. SMOTEEN improves how the model can predict all samples in the Thyroid dataset with an imbalanced ratio of 36.94 approximately 1.45, the ecoli3 dataset with an imbalanced ratio of 8.6 around 90%, the glass2 dataset with an imbalanced ratio of 11.59 around 52%, and the abalone19 dataset with an imbalanced ratio of 129.43 around 74%.

The F1 metric collects the advantages of precision and recall; it is the harmonic average of precision and recall, as shown in Table 3. It is worth mentioning that near-miss and oversampling techniques achieved the best results for the wine dataset. A near miss is achieved at 97% which improved by 14% according to the balance dataset. When calculating the mean for each resampling method, it is found that SMOTEEN achieves the best performance with a support vector machine classifier with an improvement of 36% according to without sampling, followed by SMOTE Tomek, and SMOTE is improved by around 25%, and random oversampling is improved by 24%. To summarize, SMOTTEN as a resampling technique achieved the best result with the SVM classifier for the largest dataset.

According to the logistic regression classifier, SMOTEEN recognizes an average of around 0.92 true positives according to all predictive positives. The classification of the glass2 dataset is improved by around 0.63% using SMOTE, SMOTE TOMEK, or SMOTEEN. Thyroid with IR = 129.43 and containing multiclass was enhanced by 1.9% using SMOTEEN. The balance dataset is achieved at 0.97 by using near miss and SMOTEEN, which improved by 56%. SMOTEEN is enhanced by around 31% according to the mean of all datasets, as shown in Table 4.

Table 4. shows that SMOTEEN and Near Miss achieved the best performance of the balance data; as they achieved 0.96. SMOTEEN, SMOTE TOMEK, and SMOTE improved the percentage from 0.5 to 0.71 and 0.75, respectively, according to the Glass2 dataset. The thyroid dataset achieved 0.99 using SMOTEEN. Overall, SMOTEEN is the best to predict all samples in the 8 used datasets.

Table 4 illustrates that by using SMOTEEN and a logistic regression classifier, the Thyroid dataset has an imbalanced ratio of 36.94, and multiclass improves classification by around 1.5%. The Glass2 dataset has improved by about 44%. The abalone dataset has improved by 74%. The Glass2 dataset achieved 0.69 by using SMOTE and hybrid oversampling and under sampling techniques (SMOTE Tomek, SMOTEEN).

Finally, applying the K Neighbors classifier, see Table 5. shows that using random oversampling and SMOTEEN on Abalone19 with an imbalanced ratio of 129.43 and two classes achieved 0.99. The Vehicle1 dataset with IR = 2.9 was improved by around 62% using SMOTEEN, 40% using SMOTE TOMEK, and SMOTE. According to the others, SMOTEEN has achieved the best performance, as it can find the most positive according to all predicted positives. Calculating Recall shows how well the model can predict each data sample.

Table 5. shows random oversampling for glass2 by the

Table 4 . Precision, Recall, and F1-score with Logistic Regression Classifier

Dataset	IR	without sampling	under sampling		Oversampling		combination	
			Near miss	Random UnderSampler	Random OverSampler	SMOTE	SMOTE Tomek	SMOTE ENN
Precision								
Wine	1.5	0.95	0.96	0.96	0.96	0.96	0.96	0.98
Vehicle1	2.9	0.64	0.56	0.7	0.77	0.83	0.84	0.96
Ecoli2	5.46	0.91	0.88	0.88	0.96	0.94	0.94	0.99
Balance	5.8	0.62	0.97	0.86	0.92	0.9	0.89	0.97
Ecoli3	8.6	0.87	0.69	0.88	0.94	0.94	0.94	0.99
Glass2	11.59	0.46	0.6	0.49	0.64	0.75	0.75	0.75
Thyroid	36.94	0.31	0.65	0.49	0.76	0.84	0.85	0.9
Abalone19	129.43	0.5	0.75	0.63	0.8	0.84	0.84	0.87
Mean		0.7(6)	0.77(4)	0.76(5)	0.83(3)	0.86(2)	0.86(2)	0.92(1)
Recall								
Wine	1.5	0.95	0.96	0.95	0.96	0.97	0.96	0.99
Vehicle1	2.9	0.63	0.56	0.69	0.76	0.8	0.81	0.98
Ecoli2	5.46	0.91	0.88	0.89	0.96	0.94	0.94	0.99
Balance	5.8	0.66	0.96	0.84	0.91	0.9	0.89	0.96
Ecoli3	8.6	0.79	0.7	0.88	0.93	0.94	0.93	0.99
Glass2	11.59	0.5	0.59	0.6	0.63	0.71	0.71	0.75
Thyroid	36.94	0.33	0.53	0.47	0.76	0.8	0.8	0.99
Abalone19	129.43	0.5	0.65	0.58	0.8	0.83	0.83	0.93
Mean		0.69(6)	0.75(5)	0.75(5)	0.83(4)	0.84(3)	0.85(2)	0.94(1)
F1-score								
Wine	1.5	0.95	0.96	0.95	0.96	0.96	0.96	0.98
Vehicle1	2.9	0.63	0.56	0.69	0.76	0.79	0.81	0.95
Ecoli2	5.46	0.91	0.88	0.87	0.96	0.94	0.94	0.99
Balance	5.8	0.64	0.96	0.84	0.91	0.9	0.89	0.97
Ecoli3	8.6	0.8	0.68	0.87	0.93	0.94	0.93	0.99
Glass2	11.59	0.48	0.55	0.45	0.62	0.69	0.69	0.69
Thyroid	36.94	0.32	0.51	0.44	0.75	0.79	0.79	0.82
Abalone19	129.43	0.5	0.59	0.54	0.8	0.83	0.83	0.87
Mean		0.69(5)	0.73(4)	0.73(4)	0.83(3)	0.84(2)	0.84(2)	0.91(1)

imbalanced ratio of 11.59 achieved an improvement of about 78%, and thyroid datasets by the imbalanced ratio of 36.94 achieved an improvement of approximately 96% without sampling. The abalone19 dataset can predict using random oversampling and SMOTEEN how accurate the model is by performing 96% as an improvement. The ecoli2 dataset can predict all true values using SMOTEEN. The abalone19 dataset (IR = 129.43) achieved 99%, which increased the performance compared with without sampling by about 98% in Table 5. It is calculating the mean of each resampling with each classifier, and SMOTEEN achieved the best performance. It is improved by 1.8% without sampling. The ecoil2 dataset achieved a high improvement.

From the above, SMOTEEEN achieved the best result using the KNN classifier with ecoli2, ecoil3, glass2, thyroid, and abalone19, as illustrated in Table5. SMOTEEEN achieved the best result using the Logistic Regression classifier with wine, vehicle1, ecoil3, and balance as shown in Table4. SMOTEEEN achieved the best result using the SVM classifier with vehicle 1, and balance, as shown in Table3. SMOTEEN is a hybrid technique that uses the advantages of overfitting the minority class and underfitting the majority class. Overall, determining the best sampling depends on the dataset and application used.

Table 5 . Precision, Recall, and F1-score with KNN Classifier

Dataset	IR	without sampling	under sampling		Oversampling		combination	
			Near miss	Random UnderSampler	Random OverSampler	SMOTE	SMOTE Tomek	SMOTE ENN
Precision								
Wine	1.5	0.7	0.66	0.76	0.75	0.79	0.81	0.97
Vehicle1	2.9	0.59	0.56	0.71	0.77	0.82	0.83	0.96
Ecoli2	5.46	0.91	0.88	0.89	0.94	0.95	0.97	0.99
Balance	5.8	0.61	0.36	0.74	0.84	0.84	0.85	0.97
Ecoli3	8.6	0.87	0.69	0.88	0.94	0.94	0.94	0.99
Glass2	11.59	0.65	0.6	0.69	0.93	0.91	0.91	0.96
Thyroid	36.94	0.33	0.44	0.27	0.95	0.91	0.9	0.96
Abalone19	129.43	0.5	0.77	0.78	0.99	0.96	0.96	0.99
Mean		0.69(6)	0.68(7)	0.75(5)	0.87(4)	0.88(3)	0.89(2)	0.97(1)
Recall								
Wine	1.5	0.69	0.65	0.75	0.74	0.78	0.81	0.98
Vehicle1	2.9	0.63	0.56	0.71	0.76	0.8	0.8	0.95
Ecoli2	5.46	0.91	0.88	0.88	0.94	0.95	0.95	1
Balance	5.8	0.62	0.3	0.72	0.8	0.81	0.82	0.96
Ecoli3	8.6	0.79	0.7	0.87	0.93	0.94	0.94	0.99
Glass2	11.59	0.51	0.59	0.73	0.91	0.9	0.9	0.96
Thyroid	36.94	0.34	0.48	0.41	0.95	0.9	0.9	0.94
Abalone19	129.43	0.5	0.71	0.77	0.99	0.95	0.95	0.98
Mean		0.68(5)	0.67(4)	0.76(4)	0.87(3)	0.87(3)	0.88(2)	0.97)(1)
F1-score								
Wine	1.5	0.68	0.64	0.74	0.73	0.77	0.79	0.97
Vehicle1	2.9	0.63	0.56	0.71	0.76	0.79	0.8	0.95
Ecoli2	5.46	0.91	0.88	0.88	0.94	0.95	0.96	1
Balance	5.8	0.61	0.29	0.71	0.8	0.81	0.82	0.97
Ecoli3	8.6	0.8	0.68	0.87	0.93	0.94	0.94	0.99
Glass2	11.59	0.51	0.54	0.63	0.91	0.9	0.89	0.95
Thyroid	36.94	0.33	0.41	0.31	0.94	0.9	0.89	0.95
Abalone19	129.43	0.5	0.68	0.76	0.99	0.95	0.95	0.98
Mean		0.68(6)	0.65(7)	0.74(5)	0.86(4)	0.87(3)	0.88(2)	0.97(1)

V. Limitations

The research was only executed on a determined dataset; therefore, we cannot generalize the result to be applicable for conversion prediction. It depends on the application and the datasets used. The technique of feature selection is not handled in this study as it limits the performance of the classifier by increasing the risk of underfitting or overfitting. It would be interesting to notice if feature selection is applied, which would achieve better performance with SMOTEEN. Because of time constraints, this study was not available.

VI. Conclusion and future work

Dealing with imbalanced data leads to misleading results because the distribution of the data is skewed. This study summarized the impact of class imbalance on the performance of the three classifiers on different datasets from Keel, Kaggle, and UCI in terms of Recall, precision, and F1-score. General comparison of pre-processing and classifying techniques with resampling methods for each dataset. Comparing balancing and classifying techniques on used datasets

- The balancing techniques specify the most improved classifier as KNeighbors, Logistic Regression, and Support Vector Machine, respectively.
- In most of the data sets, classifiers KNeighbors with SMOTEEN, SMOTE TOMEK, SMOTE, and Random Oversampling are ranked high. While SVM ranked 1st with Random under sampling, logistic regression achieved the best with near miss.
- Based on the analysis, the performance of the SMOTEEN (SMOTE with Edited Nearest Neighbor) technique performs well in comparison with other resampling techniques.

We look forward to doing oversampling and under sampling to an extent that does not change the format of the data, or it would be better to resort to Classification to deal with unbalanced data without resorting to sampling techniques. Applying feature selection to get better performance. It is possible to deal with the problem of many classes due to the unbalanced nature of the data.

References

[1] M. Ghasemaghaei, “Understanding the impact of big data on firm performance: The necessity of conceptually differentiating among big data characteristics,” International Journal of Information Management, vol. 57, p. 102055, 2021.

[2] T. Singh, R. Khanna, M. Kumar, et al., “Multiclass imbalanced big data classification utilizing spark cluster,” in 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–7, IEEE, 2021.

[3] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, Learning from imbalanced data sets, vol. 10. Springer, 2018.

[4] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, “A survey on addressing high-class imbalance in big data,” Journal of Big Data, vol. 5, no. 1, pp. 1–30, 2018.

[5] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," Journal of Big Data, vol. 6, no. 1, pp. 1–54, 2019.
[6] J. Brownlee, Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning. Machine Learning Mastery, 2020.
[7] A. Singh, R. K. Ranjan, and A. Tiwari, "Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms," Journal of Experimental & Theoretical Artificial Intelligence, pp. 1–28, 2021.
[8] T. M. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, "An investigation of credit card default prediction in the imbalanced datasets," IEEE Access, vol. 8, pp. 201173–201198, 2020.
[9] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," Information Sciences, vol. 513, pp. 429–441, 2020.
[10] Z. E. Abou Elassad, H. Mousannif, and H. Al Moatassime, "A proactive decision support system for predicting traffic crash events: A critical analysis of imbalanced class distribution," Knowledge-Based Systems, vol. 205, p. 106314, 2020.
[11] N. Yousefi, M. Alaghband, and I. Garibay, "A comprehensive survey on machine learning techniques and user authentication approaches for credit card fraud detection," arXiv preprint arXiv:1912.02629, 2019.
[12] A. Smiti, "When machine learning meets medical world: Current status and future challenges," Computer Science Review, vol. 37, p. 100280, 2020.
[13] S. Fotouhi, S. Asadi, and M. W. Kattan, "A comprehensive data level analysis for cancer diagnosis on imbalanced data," Journal of biomedical informatics, vol. 90, p. 103089, 2019.
[14] A. B. Parsa, H. Taghipour, S. Derrible, and A. K. Mohammadian, "Real-time accident detection: coping with imbalanced data," Accident Analysis & Prevention, vol. 129, pp. 202–210, 2019.
[15] S. Ashraf and T. Ahmed, "Machine learning shrewd approach for an imbalanced dataset conversion samples," Journal of Engineering and Technology (JET), vol. 11, no. 1, 2020.
[16] A. S. Hussein, T. Li, C. W. Yohannese, and K. Bashir, "A-smote: A new preprocessing approach for highly imbalanced datasets by improving smote," International Journal of Computational Intelligence Systems, vol. 12, no. 2, pp. 1412–1422, 2019.
[17] M. Bach, A. Werner, and M. Palt, "The proposal of undersampling method for learning from imbalanced datasets," Procedia Computer Science, vol. 159, pp. 125–134, 2019.
[18] 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, no. 1, pp. 1–47, 2020.
[19] A. Aada and S. Tiwari, "Predicting diabetes in medical datasets using machine learning techniques," Int. J. Sci. Res. Eng. Trends, vol. 5, pp. 257–267, 2019.
[20] F. Alahmari, "A comparison of resampling techniques for medical data using machine learning," Journal of Information & Knowledge Management, vol. 19, no. 01, p. 2040016, 2020.
[21] M. Khushi, K. Shaukat, T. M. Alam, I. A. Hameed, S. Uddin, S. Luo, X. Yang, and M. C. Reyes, "A comparative performance analysis of data resampling methods on imbalance medical data," IEEE Access, vol. 9, pp. 109960–109975, 2021.
[22] A. A. Reshi, I. Ashraf, F. Rustam, H. F. Shahzad, A. Mehmood, and G. S. Choi, "Diagnosis of vertebral column pathologies using concatenated resampling with machine learning algorithms," PeerJ Computer Science, vol. 7, p. e547, 2021.
[23] M. Abu-Arqoub, W. Hadi, and A. Ishtaiwi, "Acripper: A new associative classification based on ripper algorithm," Journal of Information & Knowledge Management, vol. 20, no. 01, p. 2150013, 2021.
[24] T. Debnath and T. Nakamoto, "Predicting individual perceptual scent impression from imbalanced dataset using mass spectrum of odorant molecules," Scientific reports, vol. 12, no. 1, pp. 1–9, 2022.
[25] K. Kim, "Normalized class coherence change-based knn for classification of imbalanced data," Pattern Recognition, vol. 120, p. 108126, 2021.
[26] J.-S. Lee, "Auc4. 5: Auc-based c4. 5 decision tree algorithm for imbalanced data classification," IEEE Access, vol. 7, pp. 106034–106042, 2019.
[27] K. M. Dolo and E. Mnkandla, "Modifying the smote and safe-level smote oversampling method to improve performance," in 4th International Conference on Wireless, Intelligent and Distributed Environment for Communication, pp. 47–59, Springer, 2022.
[28] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. A. Khalid, "Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset," in 2020 IEEE 16th International Conference on Control & Automation (ICCA), pp. 803–808, IEEE, 2020.
[29] R. Ghorbani and R. Ghousi, "Comparing different resampling methods in predicting students' performance using machine learning techniques," IEEE Access, vol. 8, pp. 67899–67911, 2020.
[30] H. A. A. Hamza, P. Kommers, et al., "A review of educational data mining tools & techniques," International Journal of Educationa3l Technology and Learning, vol. 3, no. 1, pp. 17–23, 2018.
[31] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, "Using machine learning to predict student difficulties from learning session data," Artificial Intelligence Review, vol. 52, no. 1, pp. 381–407, 2019.
[32] S. Maldonado, J. López, and C. Vairetti, "An alternative smote oversampling strategy for high-dimensional datasets," Applied Soft Computing, vol. 76, pp. 380–389, 2019.
[33] T. Lee, M. Kim, and S.-P. Kim, "Data augmentation effects using borderline-smote on classification of a p300-based bci," in 2020 8th International Winter Conference on Brain-Computer Interface (BCI), pp. 1–4, IEEE, 2020.
[34] N. Habib, M. Hasan, M. Reza, M. M. Rahman, et al., "Ensemble of chexnet and vgg-19 feature extractor with random forest classifier for pediatric pneumonia detection," SN Computer Science, vol. 1, no. 6, pp. 1–9, 2020.
[35] A. Puri and M. Kumar Gupta, "Improved hybrid bag-boost ensemble with k-means-smote–enn technique for handling noisy class imbalanced data," The Computer Journal, vol. 65, no. 1, pp. 124–138, 2022.
[36] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," ISPRS journal of photogrammetry and remote sensing, vol. 114, pp. 24–31, 2016.
[37] L. E. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, no. 2, p. 1883, 2009.

[38] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," Heliyon, vol. 4, no. 11, p. e00938, 2018.
[39] Y. ZHUANG, "Research on e-commerce customer churn prediction based on improved value model and xg-boost algorithm," Management Science and Engineering, vol. 12, no. 3, pp. 51–56, 2018.
[40] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, vol. 408, pp. 189–215, 2020.
[41] M. Brijain, R. Patel, M. Kushik, and K. Rana, "A survey on decision tree algorithm for classification," 2014.
[42] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied logistic regression, vol. 398. John Wiley & Sons, 2013.
[43] D. Berrar, "Bayes' theorem and naive bayes classifier," Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, vol. 403, 2018.
[44] X. Zheng, SMOTE variants for imbalanced binary classification: heart disease prediction. University of California, Los Angeles, 2020.
[45] A. Ariannezhad, A. Karimpour, X. Qin, Y.-J. Wu, and Y. Salmani, "Handling imbalanced data for real-time crash prediction: application of boosting and sampling techniques," Journal of Transportation Engineering, Part A: Systems, vol. 147, no. 3, p. 04020165, 2021.
[46] N. Kumar, A. Barthwal, D. Lohani, and D. Acharya, "Modeling iot enabled automotive system for accident detection and classification," in 2020 IEEE Sensors Applications Symposium (SAS), pp. 1–6, IEEE, 2020.
[47] P. Mehrannia, S. S. G. Bagi, B. Moshiri, and O. A. Al-Basir, "Deep representation of imbalanced spatio-temporal traffic flow data for traffic accident detection," arXiv preprint arXiv:2108.09506, 2021.
[48] Y. Zeinali and B. A. Story, "Competitive probabilistic neural network," Integrated Computer-Aided Engineering, vol. 24, no. 2, pp. 105–118, 2017.
[49] H. Li, J. Li, P.-C. Chang, and J. Sun, "Parametric prediction on default risk of chinese listed tourism companies by using random oversampling, isomap, and locally linear embeddings on imbalanced samples," International Journal of Hospitality Management, vol. 35, pp. 141–151, 2013.
[50] T. Pan, J. Zhao, W. Wu, and J. Yang, "Learning imbalanced datasets based on smote and gaussian distribution," Information Sciences, vol. 512, pp. 1214–1233, 2020.
[51] B. Liu and G. Tsoumakas, "Dealing with class imbalance in classifier chains via random undersampling," Knowledge-Based Systems, vol. 192, p. 105292, 2020.
[52] N. M. Mqadi, N. Naicker, and T. Adeliyi, "Solving misclassification of the credit card imbalance problem using near miss," Mathematical Problems in Engineering, vol. 2021, 2021.
[53] A. Fernández, M. J. del Jesus, and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets," International Journal of Approximate Reasoning, vol. 50, no. 3, pp. 561–577, 2009.