Information Sciences with Applications, Vol. 2, 2024
https://doi.org/10.61356/j.iswa.2024.213073

An Improved Model Using Oversampling Technique and Cost-Sensitive Learning for Imbalanced Data Problem

Shrouk El-Amir [1,*], and Ibrahim El-Henawy [1]

[1] Department of Computer Science, Faculty of Computer and Informatics, Zagazig University, Zagazig, 44519, Egypt; Emails: shelshazly@fci.zu.edu.eg; ielhenawy@zu.edu.eg.

* Correspondence: shelshazly@fci.zu.edu.eg.

Abstract: In today's world, classification learning is a vital task because of the advancement in technology. However, during the classification process, we found the classifiers (the traditional classification techniques) couldn't handle the imbalanced data, which means the instances (majority instances) that belong to one class are many more than the instances (minority instances) that belong to another class. The use of oversampling approaches and cost-sensitive strategies are two popular approaches for addressing the imbalanced class snag. However, the best outcomes are achieved by combining the two approaches. So, the paper's concentration is to propose an enhancement model by combining the cost-sensitive technique adapted from the entropy-based fuzzy support vector machine algorithm (EFSVM), called entropy-based fuzzy membership, and the oversampling method, and provide a comparison among imbalanced learning techniques on KEEL and UCI repositories. According to the experimental findings, our enhanced model will outperform all existing models in terms of performance.

Keywords: Classification; Imbalanced Data Problem; Support Vector Machine; Fuzzy Sets; Algorithms; Entropy.

---

1. Introduction

Nowadays, dealing with imbalanced datasets has become a key role in data mining. The imbalanced dataset problem means one class contains many more instances than the remaining classes. The proportion of minority to majority instances may be 1:1000, 1:10000, and 1:100000; in sum, the minority class has fewer instances than the majority class. This snag arose in multi-class data and in binary-class data too. As a rule, negative examples are typically referred to as the majority class, whereas positive examples are referred to as the minority class [1, 2]. Many applications that use imbalanced datasets include diagnostics, oil spill problems, the financial industry, malware prediction, anomaly identification, and spam prediction. In the case of addressing an imbalanced snag, the traditional classifiers are interested in the majority class, as mentioned in [3]. Many variables influence how well classification performs; a few of them are model structures, parameter tuning, and input features, but finding an efficient method to work with all classification problems is still difficult. Figure 1 refers to an imbalanced problem example with a ratio of 1000:1; for each one thousand negative/majority instances, there is one positive/minority instance. Negative instances are indicated in this figure with a green minus "-," and positive instances are indicated by a red plus "+.".

The figure illustrates that it is so difficult to observe positive instances. Also, it is a weary task to indicate a decision boundary for making the classes separable.

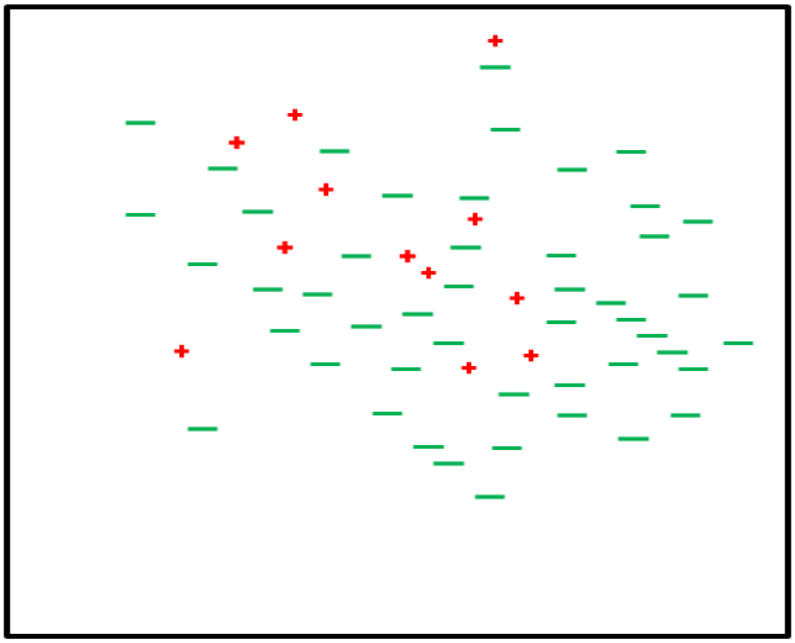

Figure 1. Imbalanced problem example.

Many solutions are given to address the imbalanced class snag, which is divided into the following categories [4]. (1) Data-level approaches that resample the instances to increase the accuracy of predictions [5, 6]. Oversampling, undersampling, and hybrid approaches are covered in this section. The first part discusses oversampling approaches, and it introduces synthetic examples into the minority class. The second subsection is called undersampling techniques, where it cuts out the original instances in the negative class to balance the dataset, while hybrid approaches combine oversampling and undersampling approaches together. (2) Algorithm-level approaches that change the immediate models to bias toward the weak class [7, 8] without adjusting the original data. (3) The cost-sensitive strategy is a mixed technique that blends data and algorithm-level strategies. This strategy provides penalties to training samples and adjusts the learning stage to receive penalties [9, 10]. The model in this category is interested in minority instances by giving higher penalties for these instances and trying to reduce the total penalty to increase the accuracy of predictions. There are different criteria that are studied to achieve acceptable performance, where accuracy is not enough for making a good decision and high accuracy does not necessarily refer to a perfect model. In the imbalanced data case, the traditional classifier will almost train on the majority of instances only in the training process, which will lead to high accuracy.

This paper strives to obtain a general insight into the imbalanced classification snag and machine learning classifiers convenient for such snags. In the research, we propose an improved model for conducting a comparative analysis with other cost-sensitive algorithms and other resampling algorithms.

This content is arranged in the following order: Regarding Section 2, sampling techniques and cost-sensitive techniques are summarized in the literature review part, and performance measures

are mentioned. Regarding Section 3, our improved model will be described. The experimental setup and findings are discussed in Part 4. In the last section, the conclusions are presented.

2. Literature Review

Researchers have offered a wide range of techniques to solve the snag of class imbalance. These techniques can be separated into data-level, algorithm-level, and cost-sensitive learning. The first solution depends on rebalancing the data sets using the resampling method to enhance accuracy. Regarding the second solution, the original classifiers are adjusted for bias towards the minor class. The third solution combines algorithm-level and data-level strategies by applying higher penalties to positive cases and decreasing these costs.

2.1 Imbalanced Problem Based On Oversampling Methods

The traditional over-sampling strategy is Random Over-Sampling (ROS) [11], which randomly reproduces minor (positive) cases from the unbalanced data set, bringing the ratio of minor (positive) to major (negative) cases close to one. Fernandez et al. [4] applied a new over-sampling strategy called Synthetic Minority Oversampling Technique (SMOTE) to avoid the drawbacks of the traditional random over-sampling strategy, where minority instances are oversampled by reproducing new instances. This algorithm concentrates on the feature space and k nearest instances for generating these new instances. Tao et al. [12] applied a novel sampling strategy in which they utilize real-valued negative selection (RNS) for producing minority instances to make imbalanced data sets more balanced and then utilize the minority instances and the majority instances as input to a classical classifier to define the optimal decision boundary as much as possible. The Borderline SMOTE strategy [13] classifies the minority instances into three categories: noise points, border points, and safe points. Then it uses the SMOTE strategy to generate new instances based only on border points. The SVM-SMOTE [14] concentrates on reproducing new instances close to borderlines by applying the SVM model in order to establish boundaries between labels. Santos et al. [15] applied a novel oversampling strategy where the K-means technique is applied to the input instances, and then clusters that have few representatives are oversampled utilizing SMOTE without consideration of the class label. The generated instance's class label is duplicated by one of the two parents who are closest to them. Last et al. [16] applied an over-sampling strategy through three steps: clustering using the k-means clustering technique; filtering using an imbalanced ratio; and oversampling using the SMOTE technique in order to rebalance the data sets. It creates a new instance that exists only in the safe area. Firstly, using k-means clustering, input instances are clustered into k. After that, a filtering step is applied, where groups that have a high imbalanced ratio (IR > 1) are selected for oversampling. Finally, SMOTE is applied to these groups to rebalance the distribution. Chang et al. [17] applied a cluster-based over-sampling approach called modified cluster-based over-sampling (MCS) for sentiment classification. MCS is divided into three steps. Firstly, similar instances are collected into clusters based on K-means. Secondly, representative instances are selected from these clusters to be oversampled based on mcs1 or mcs2. Finally, a decision tree is implemented for the training process. However, MCS is not able to identify voice data. Letteri et al. [18] proposed two novel oversampling methods called G1No and G1No Gourmet to obtain balanced data. Regarding G1No's stage, the mean and the standard deviation are calculated for each feature. Regarding G1No Gourmet's stage, the

weight for overall samples is calculated, then the mean silhouette coefficient weighted and the standard deviation silhouette coefficient weighted are calculated for each feature. The results for each method are passed to the Gaussian Random Number Generator to generate synthetic instances. Finally, perceptron is trained using this balanced data. Nam et al. [19] used a hybrid approach by developing a weighted support vector machine by applying an oversampling technique before the training process to solve the forest fire problem.

2.2 Imbalanced Problem Based On Under Sampling Methods

The traditional under-sampling technique is the Random Under-Sampling Strategy (RUS), which selects the negative instances in a random manner and erases them from the dataset until the proper distribution of classes is satisfied. Arafat et al. [20] applied an undersampling strategy by applying SVM to majority instances to select support vector decision boundaries from majority instances instead of removing parts of these instances randomly. The Edited Nearest Neighbor (ENN) technique [21] eliminates samples that are not suitable for the majority of their k-nearest neighbors. It means that this method ignores instances that can be borderline or noise. The Tomek Links [22] technique eliminates only Tomek Links from the majority of instances. Most instances are deleted by examining Tomek links between nearest-neighbor pairs. Shahabadi et al. [23] applied a clustering-based undersampling strategy by applying k-means clustering in the data pre-processing step, and then the balanced data were trained and tested using the C4.5 DT. Lee et al. [24] proposed a classification strategy via three steps. Firstly, the RUS method is applied to generate balanced data. Secondly, each instance is converted into a line graph using label encoding and discretization. Finally, a convolutional neural network is used for the training process. However, this strategy has not been tested on the multi-class problem.

2.3 Imbalanced Problem Based On Hybrid Sampling Methods

Lu et al. [25] used the SMOTE and removed the majority class instances through the ENN using the RF Classifier. Wang et al. [26] merged the SMOTE technique for generating synthetic data and the Tomek Links technique for removing some of the majority instances (Tomek links). Song et al. [27] applied a new hybrid resampling strategy where the K-means algorithm is executed on minority and majority cases in a separate way. Regarding the majority class, K-means is executed for partitioning this class into k clusters, and then the under-sampling is performed by keeping only the k nearest instances from each cluster centroid. Regarding the majority class, K-means is executed for partitioning this class into 2 partitions, and then SMOTE is executed on the smaller partition. This procedure is continued until the class size is equal. Chen et al. [28] improved a technique that merges undersampling and oversampling approaches. Firstly, the dataset is balanced by undersampling the majority instances, and then oversampling is used to enhance the diversity and data distribution, but the proposed technique parameters were left at their default values without any kind of parameter optimization strategy. Table 1 lists the advantages and disadvantages of different sampling techniques.

Table 1. Sampling methods advantages and disadvantages on imbalanced datasets.

Algorithm	Dataset	Sampling method	Advantages	Disadvantages
SMOTE	CMC, Haberman, Glass5, Glass6;	Oversampling	-Balance datasets. - No loss of information.	-Not good for high dimensional data. -Ignore neighboring instances from other classes. -May cause overlappin. -Not good in case of class noise.
RUS	CMC, Haberman, Glass5, Glass6;	Undersampling	-Balance datasets.	- Loss of information.
ROS	CMC, Haberman, Glass5, Glass6;	Oversampling	-Balance datasets.	-Cause overfitting.
SMOTE-ENN	CMC, Haberman, Glass5, Glass6;	Oversampling and Undersampling	-Eliminates the instance and its K-nearest neighbor when the class of the instance and the majority class from the instance's K-nearest neighbor are mismatched.so, it can give a deeper look in data cleaning.	-Hard to find the optimal values of k.
K-means SMOTE	CMC, Haberman, Glass5, Glass6;	Oversampling	-Avoid noise through oversampling safe instances only.	-Hard to find the optimal values of k.
SMOTETomek	CMC, Haberman, Glass5, Glass6;	Oversampling and undersampling	-Remove noise and borderlines	-Hard to find the optimal values of k

2.4 Imbalanced Problem Based On Cost Sensitive Methods

Choudhary et al. [29] suggested an algorithm that utilizes a fuzzy clustering strategy to break down the complicated imbalance challenge into sub-problems and then distributes weights to each sub-classifier for majority voting. Mienye et al. [30] developed some of the cost-sensitive algorithms by adjusting their objective functions without changing the original data distribution, but these algorithms may neglect the majority of instances in the process. Liu et al. [31] extended the work of the fuzzy SVM algorithm in the presence of borderline noise using a new method of measuring distance and the gaussian fuzzy that gave lower values for the noise and outliers to reduce their influence, but the limitation of this algorithm is that the algorithm needed more tuning. Friedman et

al. [32] applied a gradient boosting algorithm (GBM) where each control variable in gradient boosting tries to rectify the loss of its predecessor based on CART trees, where the first tree T_1 is trained using (X,Y), then the predicted labels Y′ are used to calculate the training set residual errors R_1. T_2 is then trained using (X, R_1) as labels, and so on. This procedure is carried out until all N trees have been trained. Extreme Gradient Boosting (XGBoost) [33] is based on GBM and is designed to improve the speed and performance of GBM by assigning weights to each tree.

LightGBM [34] is an improvement from the GBM, where it splits the tree leaf-wise. So it can reduce loss more than any other level-wise algorithm. It retains the samples that have large gradients and downs the samples that have small gradients in a random way using the Gradient-Based One-Side Sampling (GOSS) strategy. It also reduces the number of features using the Exclusive Feature Bundling (EFB) strategy to ameliorate efficiency while maintaining a high degree of quality. Wang et al. [35] proposed Focal-XGBoost and Weighted-XGBoost, which merge the XGBoost algorithm with focal and weighted methods to deal with imbalanced classification snags by reducing the significance of the well-classified instances. Table 2 lists the advantages and disadvantages of different cost-sensitive techniques.

Table 2. Cost-sensitive techniques advantages and disadvantages on imbalanced datasets.

Algorithm	Dataset	Advantages	Disadvantages
GBM	CMC, Haberman, Glass5, Glass6;	-Balance datasets. - Flexible.	- Cause overfitting. -More memory consumption. -More time consumption.
LightGBM	CMC, Haberman, Glass5, Glass6;	-Balance datasets. - Less memory consumption. - More fast.	- Cause overfitting sometimes.
XGBoost	CMC, Haberman, Glass5, Glass6;	-Balance datasets. -Fast.	-Cause overfitting sometimes.
Focal-XGBoost	CMC, Haberman, Glass5, Glass6;	-Balance datasets.	-More and more time consumption.
Weighted-XGBoost	CMC, Haberman, Glass5, Glass6;	-Balance datasets.	-More and more time consumption.

2.5 Performance Measures For Imbalanced Problem

We have mentioned the techniques that deal with imbalanced data sets snags. Despite the fact that accuracy and error rate are frequently utilized and simple to calculate and interpret, they have certain limitations when facing the unbalanced data sets. Wherefore, we mention the prevalent performance measures and their suitability for imbalanced data. The prevalent performance measures are driven by the confusion matrix [36], such as F-measure, precision, recall, and G-mean.

The recall rate indicates how many positive instances were successfully predicted in the sample. It is calculated as:

$$\text{recall} = \text{TP}/(\text{TP} + \text{FN}) \quad (1)$$

where *TP* (True Positives) represents the correct classification of minor(positive) instances. *FN* (False Negatives) represents positive instances misclassified as negative.

The precision rate is the percentage of correctly classified samples among the ones classified as minor (positive) . It is determined as:

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2)$$

where FP (False Positives) represent negative instances misclassified as positive.

The F-measure [37] is a weighted harmonic mean between Positive Predictive Value and True Positive Rate (TPR). The following equation illustrates the standard $\beta - \text{f} - \text{measure}$, $\text{F}-\text{measure}$ formulation, where β controls the importance of each term.

$$\text{F} - \text{measure} = \frac{\left(1+\beta^2\right)(\text{positive predictive value.sensitivity})}{\left(\beta^2.\text{positive predictive value}\right)+\text{sensitivity}} \quad (3)$$

where positive predictive value refers to precision and $sensitivity = \frac{TP}{TP+FN}$.

The Geometric mean [38] is a different form of F-measure that uses the geometric mean of precision and recall instead of using the harmonic mean. The formula for G-mean is determined as follow:

$$\text{G} - \text{mean} = \sqrt{\text{precision} \cdot \text{recall}} \quad (4)$$

3. An Improved Model

As we mentioned previously, there are three solutions that have been presented by researchers to avoid the imbalanced dataset problem. These solutions are separated into data-level, algorithm-level, and the cost-sensitive learning. In our improved model, we merge the data-level approach and cost-sensitive technique together. Algorithm 1 describes our proposed enhancement model adopted from the EFSVM algorithm [39] and the base model is Boosted Random Forest[40]. Firstly, we apply the ROS technique [11]. Then, we calculate the weights (Entropy-based Fuzzy Membership) for each sample through assigning large weights with a value of 1.0 to positive instances to guarantee their importance and fuzzified the rest of the instances according to their class certainty. The class certainty for each negative instance is determined by entropy, because entropy measures how much information is typically included in the received message [41]. The entropy for each negative instance is calculated as follows:

$$E_i = -p_{+i} ln(p_{+i}) - p_{-i} \; ln\,(p_{-i}) \quad (5)$$

where p_{+i} signifies the positive instances probability while p_{-i} signifies the negative instances probability depending on each instance's K-nearest neighbours because the local information for each instance can be provided by its neighbors (KNN) [42].

After calculating the entropy for each negative instance, we divide the negatives into subsets (m) { $sub_1, sub_2, \ldots, sub_{\text{m}}$ } through calculating the *low* and *up* values as follows:

$$Low = E_{min} + \frac{l-1}{m}\,(E_{max} - E_{min}) \quad (6)$$

$$Up = E_{min} + \frac{l}{m}(E_{max} - E_{min}) \quad (7)$$

Where E_{max} is the maximum entropy, E_{min} is the minimum entropy, m is the number of subsets and $l \in [1, m]$ [39].

After that, the fuzzy memberships of negative instances in each subset are determined as follows:

$$FM_l, = 1.0 - (\beta * (l-1)), l = 1, 2. . .. , m \quad (8)$$

where FM_l is the fuzzy membership and the fuzzy membership parameter is

$\beta \in (0, \frac{1}{(m-1)}]$ [39].

So, the Entropy-based Fuzzy Membership is determined as follows [39]:

$$W_i = \begin{cases} 1.0, & \text{if } y_i = +1 \\ FM_l, & \text{if } y_i = -1 \&\& \ x_i \in Sub_l \end{cases} \quad (9)$$

where W_i represents the fuzzy membership for instances and y_i is the actual class.

Finally, we apply Boosted Random Forest [40] to classify the balanced and weighted dataset. Figure 2 shows the main architecture of our improved model.

Algorithm 1 The Enhancement Algorithm adopted from EFSVM

1. **Begin**
2. **Input**: The training data $S = \{ x_i, y_i \}_{i=1}^{N}$, k = "nearest neighbors value", m = "subset m count", and β = "fuzzy membership value", , N_- = "number of negative samples", $y_i \in \{-1, +1\}$, $y_i = -1$: the instance x_i refers to the negative class, else refers to the positive class, E_{min}= "minimum entropy", E_{max}= "maximum entropy", E_{-i} ="negative sample entropy", num_{+i} : "number of positive instances located in KNN" , num_{-i} : " number of negative instances located in KNN" , W_i:"weight of each instance", $W_i^{(t+1)}$:"updated weights", $\propto_t$: " weighted value" ,ε_t: " error rate".
3. **Output**: Classification Model
4. **Steps:**
5. Define *No_Negative_Instances* to hold the number of negative instances
6. Define *No_Positive_Instances* to hold the number of positive instances
7. Calculate *difference= No_Positive_Instances - No_Negative_Instances*
8. i =1
9. while i ≤ *abs (difference)*
10. **Apply RandomOverSampler (x_{+i})**
11. Calculate *difference = No_Positive_Instances - No_Negative_Instances*
12. i = i+1
13. **end while**
14. Calculate the *k* value for each negative instance x_{-i}

15. Determine num_{+i} and num_{-i} for negative instance x_{-i}
16. Calculate the class probability of x_{-i}.
17. Compute the entropy for each negative instance, x_{-i} as: $E_{-i} = -p_{+i} ln(p_{+i}) - p_{-i} ln(p_{-i})$
16. **for** l ∈ 1 … m
17. calculate *Low* and *Up* using Eq. (6) and Eq. (7), respectively
19. **for** i=1 to N_
20. **if** $Low \leq H_{-i} < Up$
21. x_{-i} is located in sub$_l$
22. **end if**
23. **end for**
24. **end for**
25. Assign the fuzzy membership to each instance *i* using Eq. (9)
26. **ApplyRF(**S,F**)**
27. H = ∅
28. **for** i = 1 to *P*
29. $S^{(i)}$ ← Random instances from S, W
30. h_i ← BuildTree ($S^{(i)}$, *F*)
31. Set the content of h_i to H
32. **end for**
33. **return** H
34. **ApplyBuildTree (S,F)**
35. **for** each node
36. Select small features from F, S_T ← S(f)
37. Classifier ← $Grow_{DT\ (S_T, w_i)}$
38. **for** each leaf node
39. **if** each instances in S_T belong to one label
40. return this label
41. **else if** Attribute has no node, then return majority label
42. retrun majority label
43. **else**
44. retrun the best attribute according to the information gain
45. **end if**
46. **end for**
47. Choose this attribute with outputs .
48. Split S_T into S_1, S_2 based on the outputs
49. **end for**
50. Estimating class label: $\hat{y}_i = \arg Max_{Voting}(H, x_i)$

51. Compute error rate of classifier ε_i:

$$\varepsilon_t = \sum_{i:Yi \neq \hat{y}_i}^{N} W_i^{(t)} / \sum_{i=1}^{N} W_i^{(t)}$$

52. Compute Weight of classifier α_T:

$$\propto_t = \frac{1}{2}\log\frac{(M-1)(1-\varepsilon_t)}{\varepsilon_t}$$

53. **if** $\alpha > 0$

54 Compute $W_i^{(t+1)}$ as:

$$W_i^{(t+1)} = \begin{cases} W_i^{(t)} \exp(\propto_t), & if \quad Yi \neq \hat{y}_i \\ W_i^{(t)} \exp(-\propto_t), & otherwise, \end{cases}$$

55 **else**

56 reject a classifier

52. **end if**

53 **end**

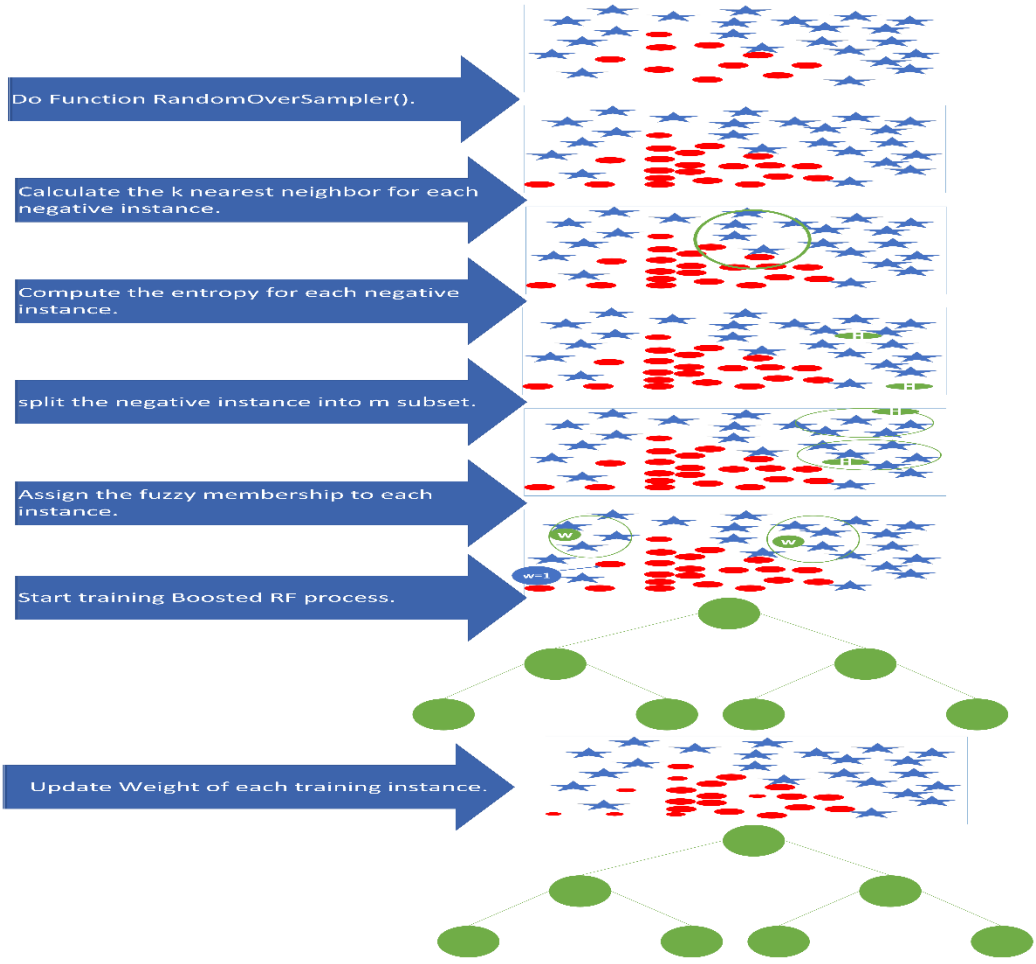

Figure 2. Main architecture of our improved model.

4. Experimental Study

This part explains the experimental study, which has been done to present the different resampling techniques' performance and cost-sensitive techniques' performance to face the imbalanced datasets utilizing RF and GBM. Firstly, we explain the classification data sets that are utilized in the experiments. Thereafter, we will explain a study about the evaluation of RF, which is blended with resampling and cost-sensitive techniques through the evaluation measures.

For testing the accuracy of the techniques presented by previous researchers for the unbalanced dataset, we determined four data sets that are accessible through the UCI Machine Learning Repository and KEEL repository. These data sets are the CMC dataset, the Haberman dataset, the Glass5 dataset, and the Glass6 dataset. Table 3 represents these datasets, where A indicates the attribute number, the dataset's size is indicated by S, and the terms "min" and "maj" designate, respectively, the cases in the dataset that belong to the minor class and the major class. CL indicates the minor class, while the term "IR" indicates the imbalanced ratio.

The experiments are executed with python language version 3.5.2 based on the original RF and GBM on Microsoft Windows 10 OS. The tests are conducted on laptop equipped with an Intel(R) Core(TM) i5-4330M processor running at 2.80GHz and 8.00G of RAM.

Table 3. KEEL and UCI dataset details.

Dataset	A	S	Min/Maj	CL	IR
Cmc	9	1473	333/1140	2	3.4234
Haberman	3	306	81/225	2	2.7778
Glass5	9	214	9/205	Positive	22.7777
Glass6	9	214	29/185	Positive	6.38

We test the standard RF, which is incorporated with the resampling techniques, by using F-measure, precision, recall, and G-mean measures. Table 4 represents different measures obtained by the RF classifier using six resampling techniques and our improved model. Each dataset's best performance is indicated by a bold font. In the CMC dataset, the highest recall goes to K-means SMOTE+RF with 0.949 and the lowest performance belongs to Original RF with 0.594 among all other techniques. Our improved model, with a G-mean of 0.966, outperforms all other techniques. Also, the highest precision belongs to our improved model, with 0.968, and the lowest one belongs to ROS+RF, with 0.389, among all other techniques. Our improved model, with an F-measure of 0.960, outperforms all other techniques. As seen in the Haberman dataset, our improved model, with a G-mean of 0.939, outperforms all other techniques. Regarding the improved model, with an F-measure of 0.928, outperforms all other techniques. Also, the highest recall in the Haberman dataset belongs to K-means SMOTE+RF with 0.845 and the lowest performance belongs to Original RF with 0.470 among all other techniques. The highest precision belongs to our improved model with 0.944, and the lowest one belongs to Original RF with 0.2 among all other techniques. In the Glass5 dataset, the highest recall goes to K-means SMOTE+RF, RUS+RF, and original RF with 1.0. It means that K-means SMOTE +RF, RUS+RF, and original RF correctly identify 100% of all different samples. The highest precision belongs to K-means SMOTE +RF, RUS+RF, and the original RF with 1.0. It means that the

original RF and K-means SMOTE +RF. RUS+RF classifies all true positive instances successfully. In the Glass6 dataset, the improved model, with a recall 1.0. It means that our improved model correctly identifies 100% of all different samples. Our improved model also has a precision of 1.0, which means that it catches all true positive instances. In the case of F-measure and G-mean, our improved model with 1.0, which means that it has a high performance in the positive instance prediction if the imbalanced ratio is high.

Table 4. Different measures obtained by RF classifier using six resampling techniques.

		Original Rf	SMOTE+ Rf	RUS+RF	ROS+RF	SMOTE-ENN+RF	K-means SMOTE+RF	SMOTETomek +RF	Improved model
CMC	f-measure	0.609	0.825	0.623	0.612	0.641	0.948	0.634	**0.960**
	G − mean	0.594	0.825	0.622	0.621	0.707	0.949	0.647	**0.966**
	Recall	0.594	0.825	0.622	0.621	0.707	**0.949**	0.647	0.733
	Precision	0.419	0.836	0.613	0.389	0.399	0.962	0.418	**0.968**
Haberman	f-measure	0.463	0.757	0.585	0.509	0.584	0.861	0.461	**0.928**
	G − mean	0.470	0.751	0.586	0.510	0.608	0.845	0.458	**0.939**
	Recall	0.470	0.751	0.586	0.510	0.608	**0.845**	0.458	0.593
	Precision	0.2	0.704	0.6	0.273	0.367	0.838	0.208	**0.944**
Glass5	f-measure	**1.0**	0.990	**1.0**	0.855	0.740	**1.0**	0.855	0.992
	G − mean	**1.0**	0.990	**1.0**	0.990	0.740	**1.0**	0.990	0.992
	Recall	**1.0**	0.990	**1.0**	0.990	0.740	**1.0**	0.990	0.990
	Precision	**1.0**	0.981	**1.0**	0.667	0.5	**1.0**	0.667	0.992
Glass6	f-measure	0.947	0.979	**1.0**	0.947	0.947	0.989	0.947	**1.0**
	G − mean	0.857	0.979	**1.0**	0.857	0.857	0.988	0.857	**1.0**
	Recall	0.857	0.979	**1.0**	0.857	0.857	0.988	0.857	**1.0**
	Precision	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.978	**1.0**	**1.0**

Because recall is very important in lots of fields, such as biomedical and bioinformatics, since these fields are related to human life. The K-means SMOTE +RF algorithm has a high recall. So, this algorithm does not miss any true positive instances. Our improved model has a high precision value, which means that it catches all true positive instances. Tables 5, 6, 7, and 8 show mean results for 30 independent runs, respectively, for the metrics we used.

Table 5. Mean recall results for cost-sensitive techniques.

Recall		GBM	LightGBM	XGBoost	Focal-XGBoost	Weighted-XGBoost	Improved Model
CMC	Average	0.590	0.601	0.594	0.613	0.623	**0.750**
	Best	0.631	0.643	0.661	0.647	0.681	0.788
	Worst	0.554	0.556	0.545	0.567	0.572	0.721
Haberman	Average	0.554	0.574	0.543	0.531	0.549	**0.661**
	Best	0.652	0.664	0.637	0.579	0.660	0.699
	Worst	0.461	0.502	0.410	0.452	0.501	0.610
Glass5	Average	0.895	0.863	0.790	0.734	0.734	**0.997**
	Best	1.0	1.0	1.0	0.75	0.75	1.0
	Worst	0.5	0.490	0.5	0.481	0.490	0.990
Glass6	Average	0.898	0.914	0.921	0.918	0.904	**0.975**
	Best	1.0	1.0	1.0	0.929	0.929	0.989
	Worst	0.714	0.786	0.786	0.836	0.847	0.946

Table 6. Mean G-mean results for cost-sensitive techniques.

G − mean		GBM	LightGBM	XGBoost	Focal-XGBoost	Weighted-XGBoost	Improved Model
CMC	Average	0.590	0.601	0.594	0.613	0.623	**0.970**
	Best	0..631	0.643	0.661	0.647	0.681	0.978
	Worst	0.54	0.556	0.545	0.567	0.572	0.956
Haberman	Average	0.554	0.574	0.543	0.531	0.549	**0.970**
	Best	0.652	0.664	0.637	0.579	0.660	0.989
	Worst	0.461	0.502	0.410	0.452	0.501	0.956
Glass5	Average	0.895	0.863	0.790	0.734	0.734	**0.998**
	Best	1.0	1.0	1.0	0.75	0.75	1.0
	Worst	0.5	0.490	0.5	0.481	0.490	0.992
Glass6	Average	0.898	0.914	0.921	0.918	0.904	**0.996**
	Best	1.0	1.0	1.0	0.929	0.929	1.0
	Worst	0.714	0.786	0.786	0.836	0.847	0.986

Table 7. Mean precision results for cost-sensitive techniques.

precision		GBM	LightGBM	XGBoost	Focal-XGBoost	Weighted-XGBoost	Improved Model
CMC	Average	0.407	0.461	0.415	0.452	0.474	**0.975**
	Best	0.492	0.563	0.538	0.533	0.557	0.981
	Worst	0.329	0.364	0.333	0.366	0.407	0.967
Haberman	Average	0.338	0.420	0.330	0.308	0.329	**0.970**
	Best	0.747	0.563	0.474	0.389	0.429	0.996
	Worst	0.188	0.263	0.111	0.176	0.261	0.952
Glass5	Average	0.763	0.756	0.783	0.628	0.628	**0.997**
	Best	1.0	1.0	1.0	1.0	1.0	1.0
	Worst	0.0	0.0	0.0	0.0	0.0	0.991
Glass6	Average	0.808	0.906	0.936	0.886	0.856	**0.994**
	Best	1.0	1.0	1.0	1.0	1.0	1.0
	Worst	0.545	0.6	0.714	0.714	0.75	0.997

Table 8. Mean f-measue results for cost-sensitive techniques

f-measue		GBM	LightGBM	XGBoost	Focal-XGBoost	Weighted-XGBoost	Improved Model
CMC	Average	0.603	0.624	0.608	0.630	0.641	**0.963**
	Best	0.652	0.667	0.680	0.671	0.686	0.976
	Worst	0.562	0.557	0.555	0.582	0.591	0.940
Haberman	Average	0.552	0.585	0.543	0.531	0.546	**0.967**
	Best	0.641	0.683	0.641	0.583	0.632	0.988
	Worst	0.455	0.490	0.404	0.447	0.500	0.952
Glass5	Average	0.882	0.863	0.804	0.778	0.778	**0.998**
	Best	1.0	1.0	1.0	0.909	0.909	1.0
	Worst	0.485	0.483	0.485	0.481	0.483	0.991
Glass6	Average	0.888	0.933	0.950	0.929	0.912	**0.996**
	Best	1.0	1.0	1.0	0.976	0.976	1.0
	Worst	0.774	0.801	0.836	0.836	0.872	0.985

Each dataset's best performance is indicated by a bold font. In the CMC dataset, the highest recall goes to our improver model with 0.750, and the lowest one goes to GBM with 0.590, among all other techniques. Our improved model, with a G-mean of 0.970, outperforms all other techniques. Also, the highest precision belongs to our improved model, with 0.975, and the lowest one belongs to GBM, with 0.407, among all other techniques. Our improved model, with an F-measure of 0.963, has the

best performance compared with other techniques. Regarding the Haberman dataset, our improved model, with a G-mean of 0.970, outperforms all other techniques. Our improved model, with an F-measure of 0.967, outperforms all other techniques. The Haberman dataset's highest recall goes to our improved model with 0.661, and the lowest one belongs to Focal-XGBoost with 0.531 among all other techniques. The highest precision belongs to our improved model with 0.970, and the lowest one belongs to Focal-XGBoost with 0.308, among all other techniques. The Glass5 dataset's highest recall goes to the improved model with 0.997, and the lowest one belongs to Focal-XGBoost and Weighted-XGBoost with 0.734, among all other techniques. The highest precision belongs to the improved model with 0.997, and the lowest performance belongs to Focal-XGBoost and Weighted-XGBoost with 0.628, among all other techniques. The improved model, with an F-measure of 0.998, outperforms all other techniques. Regarding the improved model, with a G-mean of 0.998, it outperforms all other techniques. The highest recall in the Glass6 dataset belongs to the improved model compared with other techniques, with 0.975. The highest precision belongs to the improved model, with 0.994 compared with other techniques. The improved model, with an F-measure of 0.996, outperforms all other techniques. The improved model, with a G-mean of 0.996, outperforms all other techniques. We notice that our improved model outperforms all algorithms, whether resampling algorithms or cost-sensitive algorithms, because it can catch all true positive instances.

5. Conclusions

Regarding the problems of imbalanced data sets, many traditional classification methods fall short because they were constructed to deal mostly with balanced situations. Many different solutions to the issue of class imbalance have been put forth by researchers. These solutions can be classified into three categories: data-level, algorithm-level, and cost-sensitive strategy. The first option focuses on rebalancing data sets using resampling techniques, and the last option focuses on assigning a high cost to rare samples. So, in this study, we examined several resampling methods and cost-sensitive strategies to enhance classification performance during imbalanced data set problems. We discussed the performance of different resampling techniques and cost-sensitive techniques on some imbalanced data sets in terms of F-measure, precision, recall, and G-mean. We also proposed an improved model adapted from the EFSVM. With respect to the metrics we have selected, we noticed that our improved model can catch all true positive instances, and most resampling techniques give satisfying performance in the case of a high imbalanced ratio. The experiment results confirm that if you want to catch many true positives and return an accurate result, our improved model is the best choice, especially in the event of an extremely imbalanced ratio. As a result, research in the future might concentrate on utilizing K-means SMOTE, our improved model, and other cost-sensitive techniques for other real-world imbalanced snags; determining the best values of k for K-means SMOTE +RF and our improved model; and using Big Data techniques with these strategies for imbalanced Big Data problems.

Author Contributions

All authors contributed equally to this work.

Funding

This research was conducted without external funding support.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

References

[1] He, H. and E.A. Garcia, Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 2009. 21(9): p. 1263-1284.
[2] Van Hulse, J. and T. Khoshgoftaar, Knowledge discovery from imbalanced and noisy data. Data & Knowledge Engineering, 2009. 68(12): p. 1513-1542.
[3] Kang, Q., et al., A noise-filtered under-sampling scheme for imbalanced classification. IEEE transactions on cybernetics, 2016. 47(12): p. 4263-4274.
[4] Fernández, A., et al., Learning from imbalanced data sets. Vol. 10. 2018: Springer.
[5] Chawla, N.V., et al., SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 2002. 16: p. 321-357.
[6] Le, T. and S.W. Baik, A robust framework for self-care problem identification for children with disability. Symmetry, 2019. 11(1): p. 89.
[7] Lin, Y., Y. Lee, and G. Wahba, Support vector machines for classification in nonstandard situations. Machine learning, 2002. 46(1): p. 191-202.
[8] Liu, B., Y. Ma, and C.K. Wong. Improving an association rule based classifier. in European Conference on Principles of Data Mining and Knowledge Discovery. 2000. Springer.
[9] Chawla, N.V., et al., Automatically countering imbalance and its empirical relationship to cost. Data Mining and Knowledge Discovery, 2008. 17(2): p. 225-252.
[10] Ling, C.X., V.S. Sheng, and Q. Yang, Test strategies for cost-sensitive decision trees. IEEE Transactions on Knowledge and Data Engineering, 2006. 18(8): p. 1055-1067.
[11] Batista, G.E., R.C. Prati, and M.C. Monard, A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter, 2004. 6(1): p. 20-29.
[12] Tao, X., et al., Real-value negative selection over-sampling for imbalanced data set learning. Expert Systems with Applications, 2019. 129: p. 118-134.
[13] Han, H., W.-Y. Wang, and B.-H. Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. in International conference on intelligent computing. 2005. Springer.

[14] Tang, Y., et al., SVMs modeling for highly imbalanced classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2008. 39(1): p. 281-288.
[15] Santos, M.S., et al., A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. Journal of biomedical informatics, 2015. 58: p. 49-59.
[16] Chen, Y. and R.J.C. Zhang, Research on credit card default prediction based on k-means SMOTE and BP neural network. 2021. 2021: p. 1-13.
[17] Chang, J.-R., L.-S. Chen, and L.-W. Lin, A Novel Cluster based Over-sampling Approach for Classifying Imbalanced Sentiment Data. IAENG International Journal of Computer Science, 2021. 48(4).
[18] Letteri, I., et al. Imbalanced Dataset Optimization with New Resampling Techniques. in Proceedings of SAI Intelligent Systems Conference. 2021. Springer.
[19] Nam, J.H., et al., Prediction of Forest Fire Risk for Artillery Military Training using Weighted Support Vector Machine for imbalanced data. 2022.
[20] Arafat, M.Y., et al. An under-sampling method with support vectors in multi-class imbalanced data classification. in 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA). 2019. IEEE.
[21] Wilson, D.L., Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, 1972(3): p. 408-421.
[22] Tomek, I., A generalization of the k-NN rule. IEEE Transactions on Systems, Man, and Cybernetics, 1976(2): p. 121-126.
[23] Shahabadi, M.S.E., et al., A combination of clustering-based under-sampling with ensemble methods for solving imbalanced class problem in intelligent systems. Technological Forecasting and Social Change, 2021. 169: p. 120796.
[24] Lee, Y.S. and C.C. Bang, Framework for the Classification of Imbalanced Structured Data Using Under-sampling and Convolutional Neural Network. Information Systems Frontiers, 2021: p. 1-15.
[25] Muntasir Nishat, M., et al., A comprehensive investigation of the performances of different machine learning classifiers with SMOTE-ENN oversampling technique and hyperparameter optimization for imbalanced heart failure dataset. 2022. 2022: p. 1-17.
[26] Hairani, H., A. Anggrawan, and D.J.J.I.J.o.I.V. Priyanto, Improvement Performance of the Random Forest Method on Unbalanced Diabetes Data Classification Using Smote-Tomek Link. 2023. 7(1): p. 258-264.
[27] Song, J., et al. A bi-directional sampling based on K-means method for imbalance text classification. in 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). 2016. IEEE.
[28] Chen, Z., et al., A hybrid data-level ensemble to enable learning from highly imbalanced dataset. Information Sciences, 2021. 554: p. 157-176.
[29] Choudhary, R. and S. Shukla, A clustering based ensemble of weighted kernelized extreme learning machine for class imbalance learning. Expert Systems with Applications, 2021. 164: p. 114041.
[30] Mienye, I.D. and Y. Sun, Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. Informatics in Medicine Unlocked, 2021. 25: p. 100690.
[31] Liu, J., Fuzzy support vector machine for imbalanced data with borderline noise. Fuzzy Sets and Systems, 2021. 413: p. 64-73.
[32] Friedman, J.H., Greedy function approximation: a gradient boosting machine. Annals of statistics, 2001: p. 1189-1232.
[33] Chen, T. and C. Guestrin. Xgboost: A scalable tree boosting system. in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.
[34] Ke, G., et al., Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 2017. 30.
[35] Wang, C., C. Deng, and S. Wang, Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. Pattern Recognition Letters, 2020. 136: p. 190-197.
[36] Luque, A., et al., The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, 2019. 91: p. 216-231.

[37] Batuwita, R. and V. Palade, Adjusted geometric-mean: a novel performance measure for imbalanced bioinformatics datasets learning. Journal of bioinformatics and computational biology, 2012. 10(04): p. 1250003.
[38] Barandela, R., et al., Strategies for learning in class imbalance problems. Pattern Recognition, 2003. 36(3): p. 849-851.
[39] Fan, Q., et al., Entropy-based fuzzy support vector machine for imbalanced datasets. Knowledge-Based Systems, 2017. 115: p. 87-99.
[40] Iwendi, C., et al., COVID-19 patient health prediction using boosted random forest algorithm. Frontiers in public health, 2020. 8: p. 357.
[41] Shannon, C.E., A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review, 2001. 5(1): p. 3-55.
[42] Qin, Z., et al. Cost-sensitive classification with k-nearest neighbors. in International Conference on Knowledge Science, Engineering and Management. 2013. Springer.

Received: 01 Sep 2023, **Revised:** 13 Jan 2024,
Accepted: 11 Feb 2024, **Available online:** 16 Mar 2024.