



Beyond the hypes on data rebalancing in imbalance learning: towards a balanced framework and recommender system

Rafiu Mope Isiaka¹ · Kabir Opeyemi Olorede² · Ronke Seyi Babatunde¹ · Falilat Jumoke Ajao¹

Received: 21 October 2024 / Accepted: 21 July 2025
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2025

Abstract

Class-imbalanced learning presents critical challenges in machine learning, largely because data in most domains are naturally imbalanced. Although resampling techniques have been widely applied to address this issue, their effectiveness has been inconsistent and sometimes flawed, owing to artificial assumptions. In this study, we move beyond hype surrounding resampling methods by exploring alternative strategies, such as ensemble learning, cost-sensitive algorithms, and one-class classification techniques. Through rigorous experimentation across extreme, moderate, and mild imbalance levels, our findings reveal that these alternatives often outperform traditional resampling in terms of F1-scores, with ensemble SVM and one-class logistic regression achieving notable values of 1.00 and 0.90, respectively. In addition, we introduce a knowledge-based recommender system designed to help practitioners choose the most appropriate techniques for addressing class imbalance. This research argues that resampling is not always the optimal solution for all instances, thereby advocating a more balanced framework that leverages advanced methods for superior performance in imbalanced learning tasks. Our study advances the field by offering a pragmatic, data-driven approach to overcoming class imbalances, contributing valuable insights for both researchers and practitioners.

Keywords Class-imbalanced learning · Resampling techniques · Ensemble learning · Cost-sensitive learning · One-class classification · Recommender system

1 Introduction

Class-imbalanced learning poses a significant challenge in machine learning and data mining, where majority class samples overwhelmingly outnumber minority class samples. In real-world applications such as fraud detection, medical diagnosis, and anomaly detection, minority classes are of greater importance, despite being underrepresented. Dataset D , with m instances across n classes $C_1, C_2, \dots, C_n$ is considered class-imbalanced when there exists a significant disparity between class sizes, such as $|C_i| \gg |C_j|$, where i and j are indexes of the classes [1]. This imbalance creates challenges for traditional machine learning models, which typically

assume balanced class distributions, leading to biased predictions and poor generalisation, particularly for minority classes [2]. It has been recognised as one of the top ten challenges in data mining and pattern recognition [3]. Imbalanced datasets can arise intrinsically, as in cases of rare diseases, or extrinsically, due to sampling methods or instrumentation errors [4]. In such scenarios, the minority class typically carries greater significance, owing to the higher cost associated with its misclassification [5]. Class-imbalanced severity is typically classified into three categories: mild (20–40%), moderate (1–20%), and extreme ($\leq 1\%$) [6].

Traditional machine learning classifiers struggle with imbalanced data because they are designed to maximise the overall accuracy, which can lead to a misleadingly high accuracy rate when the model simply predicts the majority class. This issue, referred to as the ‘accuracy paradox’, fails to accurately represent the effectiveness of the model in identifying instances of the minority class [7]. To address these challenges, researchers have introduced a range of techniques using data, algorithmic, and hybrid approaches [8].

✉ Kabir Opeyemi Olorede
kabir.olorede@kwasu.edu.ng

¹ Department of Computer Science, Faculty of Information and Communication Technology, Kwara State University, Malete, Nigeria

² Department of Mathematics and Statistics, Faculty of Pure and Applied Sciences, Kwara State University, Malete, Nigeria

The data-level approach, encompassing methods such as undersampling, oversampling, and hybrid techniques, is most commonly employed because of its simplicity and compatibility with standard machine learning algorithms [9]. These techniques attempt to rebalance the dataset to achieve a 50:50 class distribution that is suitable for classifiers. The effectiveness of these methods has been demonstrated in numerous studies using decision trees, naïve Bayes, and convolutional neural networks [10, 11]. Comparative studies of classifiers such as C5.0, multilayer perceptron (MLPs), and support vector machines (SVMs) have demonstrated that some models exhibit reduced sensitivity to sampling techniques [12–14]. The MLPs and SVMs were demonstrated by Japkowicz and Stephen [15] to be less sensitive to sampling variations. Similarly, Khushi et al. compared the performances of SVMs, logistic regression, and random forests (RFs) by resampling imbalanced medical data for classification. Also traceable in the literature is some works on imbalance learning frameworks, as in [5], Zhao, Wong [16], Zhang, Zou [17], and Jiang [18].

Resampling techniques remain a practical choice in domains that require high interpretability, such as healthcare and finance. For instance, using SMOTE in logistic regression models helps address class imbalance without compromising the explanatory power of predictor variables [19]. Similarly, resampling combined with geometric transformations effectively enhances model robustness in tasks involving limited labelled data, as demonstrated in convolutional neural networks for image classification tasks [20]. These scenarios underscore that while advanced methodologies may surpass resampling in predictive performance, resampling retains its utility under certain conditions, especially when data augmentation or interpretability is critical.

Despite their popularity, data-level techniques are limited. Oversampling can introduce bias by generating synthetic data that may not accurately represent the true distribution of test data. This can result in overfitting and exacerbating any underlying anomalies in the data. Conversely, undersampling may lead to the loss of important data, potentially distorting the effectiveness of the model and causing miscalibration [21]. Consequently, models trained on rebalanced data may face challenges in generalising real-world imbalanced datasets, requiring retraining on properly distributed test data [22]. Given these limitations, there is growing interest in exploring algorithm-level solutions that do not rely on data rebalancing. These methods include the creation of specialised algorithms, ensemble learning techniques, cost-sensitive learning, and one-class classification, which are designed to address class imbalances more effectively and ethically.

However, although significant work has been conducted on resampling techniques and algorithm-level methods, there is a need for comprehensive experimental validation across

diverse types of classifiers and imbalance levels. Previous studies have focussed on demonstrating the effectiveness of rebalancing techniques without adequately comparing them with baseline classifiers, leading to potentially exaggerated claims regarding their performance. Additionally, research on cost-sensitive and one-class learning has often concentrated on individual techniques, with little cross-referencing, to provide a broader understanding of their relative strengths and weaknesses.

This study aims to address these gaps by experimentally evaluating the performance of four classifiers, along with their corresponding ensemble, cost-sensitive, and one-class learning algorithms, across datasets with extreme, moderate, and mild imbalances, using synthesised and benchmark datasets. We also propose a rule-based recommender system for imbalanced learning that guides users in choosing the most suitable techniques for their specific needs, minimising reliance on trial and error. Our findings contribute to the advancement of imbalanced learning by providing empirical evidence that cautions the current emphasis on data rebalancing and advocates for a more balanced framework.

2 Related works

This section reviews relevant studies on data-level rebalancing techniques and algorithm-level methods, including ensemble learning, cost-sensitive learning, and one-class classification, as well as the development of imbalanced learning frameworks.

2.1 Resampling techniques

Resampling techniques are pivotal for addressing imbalanced class distributions in datasets, which is a common challenge in imbalanced learning. These techniques aim to create a more balanced dataset by oversampling the minority class or undersampling the majority class. Common resampling methods include random oversampling (ROS), random undersampling (RUS), NearMiss, and synthetic minority oversampling technique (SMOTE). The primary objective of these methods is to improve the performance of classification algorithms in accurately identifying minority class samples.

For instance, Thanka, Ram [23] demonstrated the effectiveness of resampling methods in predicting stroke incidence using three machine learning algorithms and an artificial neural network (ANN) model. The dataset consists of seventeen (17) predictor variables, achieved a high accuracy of 99.3% when resampled using a combination of SMOTE and RUS. However, the researchers acknowledged that the ANN model performed well on both original and

resampled datasets, thereby putting a question mark on the essence of the resampling.

Conversely, [24] highlighted concerns regarding oversampling, noting that it could lead to models trained on synthetic data that may not perform as expected in real-world applications. In their study, seventy (70) oversampling methods were investigated and ranked, based on the number of incorrect examples generated. The findings indicate that most oversampling methods created minority samples that were more likely to belong to the majority class, contradicting the intended purpose. The study suggested that oversampling in its current form is unreliable for real-world applications, particularly in sensitive domains such as security, autonomous driving, aviation safety, and medical applications. Instead, the authors recommend exploring ensemble approaches.

Similarly, [4] emphasised the need for rigorous evaluation of the effectiveness of undersampling and oversampling methods. Their work assessed the impact of combining seven sampling methods with eight machine learning classifiers across thirty-one (31) datasets with varying degrees of imbalance. This study found that resampling had only a minor effect on the performance of most classifiers, underscoring the need for further research to evaluate the generalisability of these techniques on larger and more diverse datasets.

2.2 Ensemble techniques

Ensemble learning methods capitalise on the combined power of multiple learning algorithms to develop high-performance, imbalanced classification models. These techniques have proved to outperform individual algorithms by enhancing the diversity of the base classifiers [25]. Shah, Gandhi [26] highlighted the benefits of ensemble learning by demonstrating how bagging, boosting, and stacking ensemble techniques can improve the predictive performance. An and Xu [27] further advanced this field by proposing a selective evolutionary heterogeneous ensemble (SEHE) algorithm. This algorithm generates diversity through various sampling strategies and trains multiple heterogeneous base classifiers. The experimental results on forty-two (42) imbalanced datasets confirmed their superiority and effectiveness.

Zhu, Zhang [28] employed ensemble learning models, such as random forest, AdaBoost, and LightGBM, to achieve high predictive accuracy. The evaluation indices demonstrated that ensemble learning algorithms consistently outperformed individual methods, affirming their feasibility for prediction tasks. However, these studies did not fully explore the significance of the higher performance of ensemble methods compared with their constituent base algorithms.

2.3 Cost-sensitive analysis techniques

Cost-sensitive learning techniques address imbalanced datasets by maintaining the natural form of the dataset and modifying the objective function of algorithms to prioritise the accurate prediction of the minority class. This approach assigns different misclassification costs to each class, emphasising the minority class through cost, probabilistic estimation, thresholds, and ensemble strategies [29].

This technique has proved effective in improving the classification performance over traditional methods, which assign a constant higher penalty for all minority instances [30]. For example, Awasthi, Sajwan [31] compared SMOTE-NC, SMOTE-ENC, and a cost-sensitive ANN (artificial neural network) to mitigate data imbalance. The study concluded that the cost-sensitive ANN technique generated superior results across versatile datasets.

Additionally, Mienye and Sun [32] developed robust cost-sensitive classifiers by modifying the objective functions of well-known algorithms such as logistic regression, decision tree, extreme gradient boosting, and random forest. These modified algorithms have been successfully applied to medical datasets, such as Pima Indians Diabetes, Haberman Breast Cancer, cervical cancer risk factors, and chronic kidney disease, yielding superior performance compared to standard algorithms.

2.4 One-class classification techniques

One-class classification (OCC) techniques address the challenge of learning from imbalanced datasets, without generating synthetic minority data. This approach eliminates bias by exclusively learning from a single class, focussing on distinguishing between a single positive class and unknown outliers.

OCC is a special case of multiclass classification, where the training data comprise only a single positive class. The goal is to accurately represent this single class while distinguishing it from the unseen outliers during exploitation. Hayashi and Fujita [33] demonstrated this approach by training one-class classifiers separately on minority and majority class data. The classifiers were then applied to compute the minority and majority scores of the testing data. Their experiments using imbalanced-learn datasets revealed that OCC methods outperformed traditional sampling methods across twenty (20) datasets when applied with decision tree and K-nearest neighbour classifiers.

Although one-class logistic regression has shown strong classification performance, its scalability depends on both feature dimensionality and noise. High-dimensional data increase computational cost ($O(n \cdot d)$ per iteration for n samples, d features) and risk overfitting in noisy settings. To mitigate this, practitioners often employ dimensionality

reduction (e.g. PCA) or feature selection before training. Chen, Yang [34] demonstrate that complexity-aware preprocessing—assessing feature overlap, sparsity, and noise prior to modelling—enhances performance in imbalanced scenarios. For noisy, high-dimensional data, hybrid pipelines that include noise filtering, feature reduction, balanced resampling (SMOTE/ADASYN), and regularised one-class logistic regression are more effective than the standalone methods. Ensemble versions, such as balanced logistic regression boosters, further increase the robustness and scalability.

2.5 Imbalanced data classification framework

Significant efforts have been made to develop imbalanced data classification frameworks that provide systematic procedures, guidelines, and automated solutions for selecting the optimal techniques. Notable frameworks include fuzzy ARTMAP, a brain-neuron framework combined with an ensemble of a specialised artificial neural network (ANN) and fuzzy C-means clustering algorithm [35]. Another prominent framework is based on federated learning, which combines deep learning with data augmentation to improve the accuracy and generalisation [36].

[5] introduced a framework for imbalanced dataset classification that begins with sampling, followed by basic classifiers and dynamic selection of classifiers. The best possible configuration was determined using a geometric mean metric (G-mean) to evaluate the dataset.

Furthermore, [16] developed a framework for learning from healthcare data with imbalanced distribution, incorporating various rebalancing strategies, and providing a guideline procedure. This two-stage framework consists of a base classifier selection stage and rebalancing strategy stage, with the recall metric used to determine correctly classified minority instances at the base classifier selection stage.

Similarly, Wu, Zhao [37] proposed a model-agnostic framework for class-imbalanced fault diagnosis. This framework addresses the challenge of acquiring failure data from large-scale equipment, which can lead to issues such as minority class overfitting and majority class domination. The framework comprises four submodules: time-series data augmentation, data-rebalanced sampler, balanced margin loss, and a classifier with a dynamic decision boundary. Experiments on public datasets demonstrated that the proposed framework outperformed various state-of-the-art methods.

3 Materials and methods

This paper presents an imbalanced learning framework that integrates basic resampling techniques, ensemble methods, cost-sensitive approaches, and one-class classification techniques. The effectiveness of these techniques was evaluated

using standard imbalanced learning metrics, such as precision, recall, and F1-score. A rule-based advisory system was developed to guide users in selecting the most appropriate imbalanced learning techniques for specific projects. The following subsections detail the development of the framework and implementation process.

3.1 Structured imbalanced learning framework

The imbalanced learning recommender framework in Fig. 1 consists of seven integrated modules designed to facilitate user decision-making based on the characteristics of the given imbalanced datasets, the goals, and the project's performance requirements.

3.1.1 Framework overview

The framework for handling class imbalance consists of six phases. First is the data analysis and preprocessing phase, the data quality is assessed, with noise, missing values, and outliers identified and cleaned. Next is the degree of imbalance checking phase, the dataset's class imbalance is determined (mild, moderate, or extreme), and balanced datasets are generated using techniques such as random oversampling, SMOTE, undersampling, and NearMiss. The algorithm selection and configuration phase involves choosing four base classifiers (logistic regression, decision tree, support vector machine, and artificial neural network), their ensembles, cost-sensitive, and one-class learning techniques with hyperparameter tuning to optimise performance. During model training, evaluation, and validation, the selected models were trained and validated using the original and rebalanced datasets by applying the precision, recall, and F1-score metrics.

The performance visualisation and interpretation phase presents the results via tables, charts, and graphs for easy interpretation. The recommendation engine and user guidance phase suggest the best imbalanced learning techniques based on dataset characteristics and user needs. Finally, the feedback and continuous improvement phase ensures that the user feedback and performance data on new datasets contribute to the ongoing refinement of the framework.

3.2 Imbalanced data curation and processing

This study used both synthetic and benchmarked imbalanced datasets in the mild, moderate, and extreme categories. These class distribution categories were used in practice to provide an intuitive understanding of the level of disparity between classes. Therefore, in this study, the categories represent the three base buckets for exploring the effects of the severity of imbalances on the distinct types of class-imbalanced learning techniques considered.

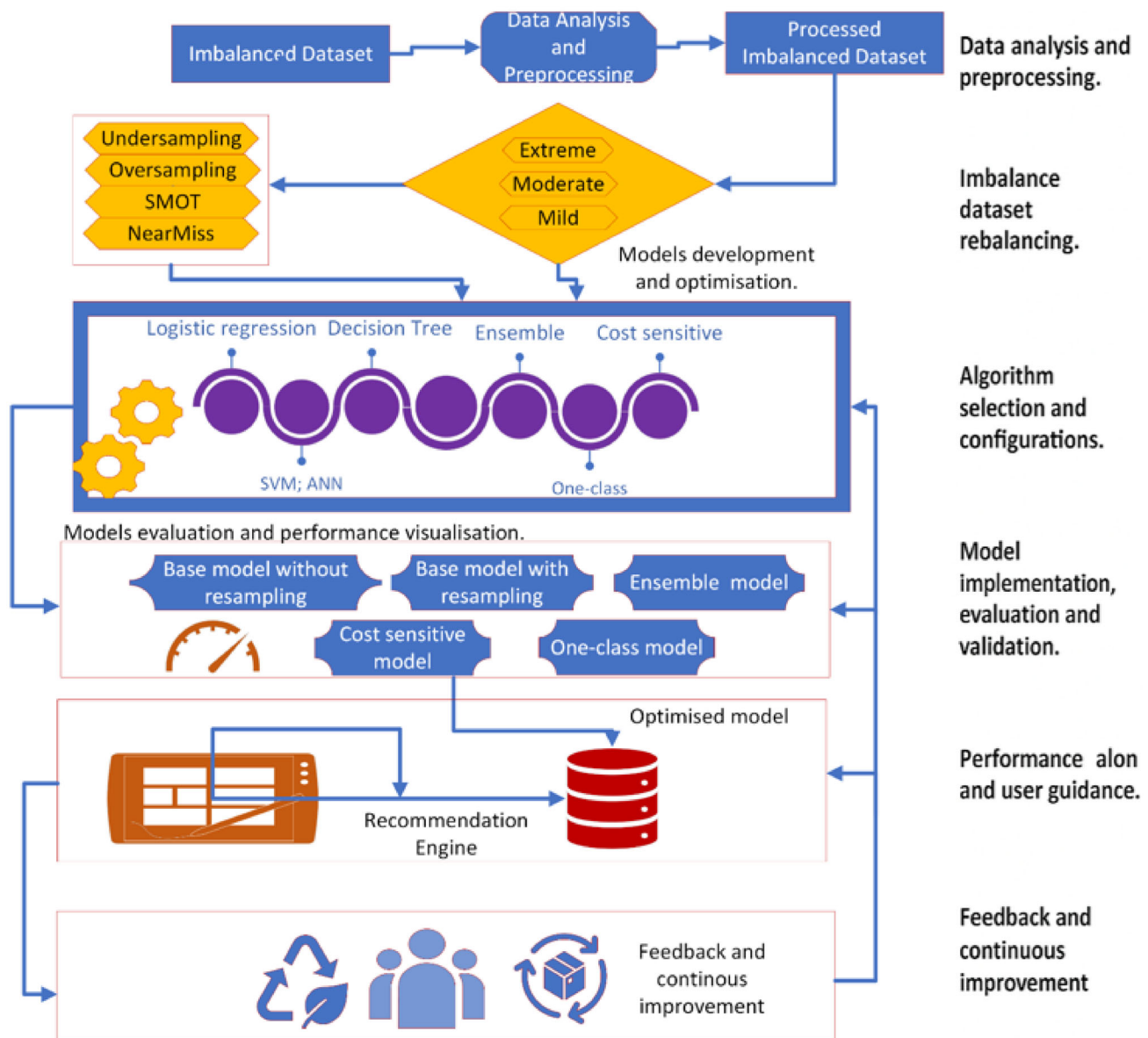


Fig. 1 Schematic diagram for imbalance learning recommender framework

The synthetic class-imbalanced dataset from the *sklearn Make_Classification* function (reference) was used to formulate the three levels of data imbalance. The weight parameters of the function were set to 0.7, 0.3, 0.9, 0.1, and 0.995, 0.005 to reflect mild, moderate, and extreme conditions, respectively.

Furthermore, three benchmark datasets from the Imbalanced-Learn (imblearn) open-source library that met mild, extreme, and moderate imbalanced conditions were chosen. The datasets according to their proportions, as shown in Table 1, are the customer churn, stroke, and credit card datasets.

The customer churn dataset is used to predict the number of customer churns from the service of a telecommunication

company. The proportion of churned customers is lower than that of non-churned customers; hence, the dataset is mildly imbalanced (36.12%). The stroke dataset contained a moderately imbalanced dataset (5.12%) in the medical rare-disease category. There is a far lower proportion of patients who are likely to experience stroke than those who are not at risk of stroke. Similarly, the credit card dataset is a fraudulent transaction dataset in which the occurrence of fraud is rare in proportion to non-fraudulent transactions, thereby making it an extremely imbalanced dataset (0.17%).

Table 1 Benchmark imbalanced dataset

Dataset	Features	Degree of imbalance	Proportion (%)	Minority	Majority	Total
Customer churn	21	Mild ($> 20\%$ $ C < = 40\%$)	36.12	1869	5,174	7,043
Stroke	11	Moderate ($> 1\%$ $ C < = 20\%$)	5.12	249	4861	5110
Credit card	31	Extreme ($ C < = 1\%$)	0.17	492	284,315	284,807

3.3 Implementation of the imbalanced learning techniques

The base imbalance learning techniques deployed in the framework include logistic regression, decision tree, support vector machines, and artificial neural networks. They were chosen to represent the linear, tree, and network types of the classifiers. Details of the implementation of these base techniques and their corresponding ensemble, cost-sensitive, and one-class techniques are presented in the following subsections.

3.3.1 Base classifiers

Four standard classifiers were used as the foundation for the ensemble, cost-sensitive, and one-class models.

Binary logistic regression: Let (X, y) define a set of N feature vectors with classification $y_i \in \{0, 1\}$. Without loss of generality, assume that X is modified as a design matrix with first column consisting of ones, representing the intercept term in the regression model. Linear logistic regression models the log-likelihood ratio of the response as a linear combination [38]:

$$\log \frac{P_r(y_i = 1|X = x_i)}{P_r(y_i = 0|X = x_i)} = \beta_0 + \beta_{xi}^T, \quad (1)$$

where $\beta_0 \in \mathbb{R}$ is the intercept term, $\beta \in \mathbb{R}^p$ is a vector of regression coefficients, and $X = (X_1, X_2, \dots, X_p)$ is a vector of predictors. Inverting this logit transformation (1) yields the conditional probability:

$$P_r(y_i = 1|X = x_i) = \frac{e^{\beta_0 + \beta_{xi}^T}}{1 + e^{\beta_0 + \beta_{xi}^T}}. \quad (2)$$

The LR, in its simplest form, does not work on classes $\{0, 1\}$ but rather the probabilities that a feature vector x_i will map to Class 1. We typically fit binary logistic models by mimicking the binomial log-likelihood of data [38].

The sigmoid (logistic) function : $S(z) = \frac{1}{1 + e^{-z}}, \quad (3)$

is an ‘S’ shaped curve with the slope parameter $\beta = 1$ that approaches 1 as $x \rightarrow \infty$ and 0 as $x \rightarrow -\infty$. It is used as an activation function to map the output of a linear function to a probability score, which can then be thresholded to make binary classification decisions.

Given the regression coefficient vector $w = (X^T X)^{-1} X^T y \in \mathbb{R}^n$ and the feature x_i , then the logistic model assumes that p_i (the probability that $y_i = 1$) is given by:

$$P_r(y_i|x_i) = p_i \sim S(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}}. \quad (4)$$

Hence the logistic classification of y_{n+1} , given unknown sample x_{n+1} is defined as:

$$y_{n+1} = \begin{cases} 1 & \frac{1}{1+e^{-w^T x_{n+1}}} > \frac{1}{2} \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

Hence logistic classifiers are a special form of linear classifiers since the definition of p_i in Eq. (4) implies that $p_i > \frac{1}{2}$ exactly when $w^T x_i > 0$.

Decision tree (DT): DT recursively splits a dataset based on feature thresholds to create a tree-like structure for classification. They make decisions by traversing from the root to the leaf nodes, where each node represents a feature and each branch represents a decision based on that feature. Techniques for handling imbalanced datasets in DT include weighted loss functions, pruning, and minimum samples per leaf. The decision tree parameters include the criterion and the maximum depth. The criterion parameter measures the quality of a split, whereas the maximum depth specifies the tree height.

Support vector machines (SVMs): SVMs are powerful supervised learning models used for classification and regression. They work by determining the optimal hyperplane that maximises the margin between classes in the feature space. They can handle both linear and nonlinear classifications by using kernel functions. The hyperparameters and settings in SVM include the selection of appropriate kernel functions, such as linear, polynomial, radial basis function (RBF), and

sigmoid. Class weights and cost-sensitive learning are techniques for handling imbalanced datasets in SVM [39].

Artificial neural network (ANN): An ANN is a composition of interconnected nodes arranged in layers. It learns complex patterns through weighted connections between nodes in different layers, and adjusts these weights during training. Hyperparameter settings include the network architecture and activation function. Adjustment of the number of layers and nodes in the network can influence the capacity of the model to learn from imbalanced data.

The handling of imbalanced datasets in an ANN is through weight initialisation, which can mitigate the influence of the majority class. Another technique is to utilise sampling methods in conjunction with ANNs to balance class distribution [40].

3.3.2 Ensemble methods for imbalanced learning

Ensemble techniques combine multiple base learners to create a more robust and effective model that outperforms the individual models. Common ensemble methods include the bagging, boosting, and classifier ensembles. Bagging (bootstrap aggregating) trains models on various subsets of the dataset, generates them through bootstrapping, and aggregates their predictions to enhance the overall performance. Boosting focuses on iteratively correcting misclassified instances, particularly those from the minority class, in subsequent models, thereby progressively improving accuracy. Examples include random forests, bagging logistic regression, and bagging SVMs, which are ensembles of decision trees, logistic regression models, and support vector machines, respectively.

Ensemble logistic regression: Ensemble linear regression typically uses bagging or stacking techniques to combine predictions from multiple binary logistic regression models trained on bootstrapped subsets of a training dataset. In Python's *sklearn* [41], the *BaggingClassifier* can be utilised to implement ensemble logistic regression with key hyperparameters including *random_state*, *solver*, *n_estimators*, and *class_weight*.

Random forest (RF): RF is an ensemble of DT that is created based on the idea of bagging. The meta-estimator employs multiple decision trees, each trained independently on random subsets of training data and random subsets of features at each node split. For classification tasks, the predictions from all trees were combined through majority voting, whereas for regression tasks, the final output was the average of the predictions. Key hyperparameters for the *sklearn* ensemble *Random Forest Classifier* include *n_estimators*, *criterion*, *max_depth*, and *min_samples_split*.

Ensemble SVM: The basic principle of ensemble SVM is similar to that of other ensemble methods, where multiple SVM models are trained using bootstrapped subsets of

the training dataset. The predictions from each model were aggregated by averaging to obtain a stronger final output. The *base_estimator* for *ensemble_svc* is (support vector classifier), and the other hyperparameters and settings for the *BaggingClassifier* include *n_estimators* and *random_state* [41].

Neural network ensembles: Ensemble neural networks can be achieved by varying the training data, models, and combinations. However, the training of neural networks is computationally expensive. A deep network trained on millions of examples may require days, weeks, or even months to train. Hence, the implementation of an ensemble neural network was exempted in this experiment [42].

3.3.3 The cost-sensitive learning methods for imbalanced learning

Cost-sensitive learning (CSL) addresses imbalanced classification problems by assigning varying costs to errors based on class distribution. By assigning higher weights to the minority class in the cost function, the model faces greater penalties for the misclassification of this class, thereby focussing more on the underrepresented class. CSL methods include cost-sensitive logistic regression, decision trees, support vector machines, and neural networks [43].

Cost-sensitive logistic regression: Cost-sensitive logistic regression, also referred to as weighted logistic regression, is an adaptation of the standard logistic regression algorithm, tailored for imbalanced datasets. This method reduces the penalty for errors in the majority class, while increasing the penalty for errors in the minority class. The mathematical expression for the cost-sensitive logistic regression model is as follows:

$$\text{logit}(p(y = 1|x)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p = \sum_{j=0}^p \beta_j x_j \quad (6)$$

where $p(y = 1|x)$ is the probability of the positive class given the input features x , β_j is the coefficient for feature x_j , and p is the number of features. The logit function is defined as $\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$.

In scikit-learn, the module for cost-sensitive logistic regression is *BalancedLogisticRegression*, where the weight for each class is specified using the *class_weight* parameter [41].

Cost-sensitive decision trees: Cost-sensitive decision trees address the issue of skewed class distributions in imbalanced learning problems. The criterion used to evaluate split points in the standard decision trees was adjusted to consider the importance of each class *during the tree construction process*. A weighted split point or weighted decision tree

typically assigns a greater weight to the minority class during training. The cost-sensitive decision tree model can be mathematically expressed as follows:

$$\text{split}(S) = \operatorname{argmax} \in S \{ \text{Gain}(S, s) - \lambda \text{Cost}(S, s) \} \quad (7)$$

$$\text{Gain}(S, s) = \text{Impurity}(S) - \sum_{t \in \{L, R\}} \frac{|S_t|}{|S|} \text{Impurity}(S_t) \quad (8)$$

$$\text{Cost}(S, s) = \sum_{i \in S} C_i w_i \quad (9)$$

where S is the set of examples at a given node, s is a candidate split point, $\text{split}(S)$ is the optimal split point, $\text{Gain}(S, s)$ is the information gain of the split, $\text{Impurity}(S)$ is the impurity of the set S , $\text{Cost}(S, s)$ is the cost of the split, C_i is the misclassification cost of example i , w_i is the weight of example i , and λ is a regularisation parameter and controls the trade-off between information gain and cost.

Balanced random forest in scikit-learn is a variant of random forest model that adapts the tree-building process to balance class distribution in each tree node and thereby reducing the impact of class imbalance.

Cost-sensitive support vector machines (SVM): In cost-sensitive SVMs, the standard SVM algorithm is modified in a way to allow the margin used to separate the classes to take the importance of each class into account. The margin can be made wider for the minority class and narrower for the majority class. The cost-sensitive SVM model can be mathematically expressed as

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C + \sum_{i: y_i = 1} \xi_i + C - \sum_{i: y_i = -1} \xi_i \quad (10)$$

Subject to $y_i (W^T x_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$.

Here, w represents the weight vector, b is the bias term, ξ_i denotes the slack variable for the i th training example, y_i is the class label for the i th training example, x_i is the corresponding feature vector. The cost parameters C_+ and C_- are associated with the positive and negative classes, respectively, and can be adjusted to reflect the relative importance of each class set to different values to reflect the importance of each class. The balanced SVM in scikit-learn is achieved by setting the *class_weight* hyperparameter to 'balanced' in addition to specification of the kernel types.

Cost-sensitive neural networks: Cost-sensitive neural networks are a variation of standard neural networks that allow the model to focus more on the minority class in datasets with severely skewed class distributions. The training algorithm modifies the backpropagation process to weigh misclassification errors according to the importance of each class, which

is often referred to as weighted or cost-sensitive neural networks.

The cost-sensitive neural network model can be mathematically expressed as follows:

$$\text{output} = f \left(\sum_{i=1}^n w_i x_i + b \right) \quad (11)$$

$$\text{loss} = \sum_{i=1}^m C_i w_i l(y_i, \text{output}_i) \quad (12)$$

where x_i is the input feature vector for the i th training example, y_i is the true class label for the i th training example, output_i is the predicted class label for the i th training example, w_i is the weight assigned to the i th training examples, b is the bias term, f is the activation function, l is the loss function, and C_i is the misclassification cost of example i . The weights for each class can be specified using the *class_weight* parameter in scikit-learn's *NeuralNetwork* class. A grid search was used to determine the optimal class weights for a given dataset.

In scikit-learn's *NeuralNetwork* class, the weights for each class can be specified using the *class_weight* parameter. Grid search can be used to find the optimal class weights for a given dataset. The *BalancedNN* in scikit-learn requires the *compute_class_weight* hyperparameter to be set to 'balanced'. Other settings included classes, *input_dim*, and activation [41].

3.3.4 One-class classification methods for imbalanced learning

One-class classification (OCC) is an unsupervised learning algorithm that focuses on modelling a single class within a dataset, eliminating bias towards other classes. It is commonly used for outlier and anomaly detection, making it a practical approach to imbalanced learning. OCC fits a model to the majority class data to determine whether a new instance belongs to the majority class (the normal group) or the minority class (the anomaly group). In this study, the one-class classification algorithms used include one-class logistic regression, isolation forest, one-class support vector machines, and one-class neural networks.

One-class logistic regression: One-class logistic regression is a binary classification model that is commonly employed for anomaly detection. It models the relationship between a binary dependent variable and one or more independent variables. The dependent variables take values of either 0 or 1, whereas the independent variables can be either continuous or categorical.

Isolation forest: Isolation forest [44] is an unsupervised learning algorithm that works by randomly partitioning the

data into subsets and recursively splitting each subset until every data point is isolated in its own subset. The key intuition is that outliers with feature values significantly different from the majority are likely to be isolated with fewer splits compared to normal points. As a result, the path length from the root node to the leaf node in an isolation tree serves as a measure of how anomalous a point is. The average path length across all trees in the forest, known as the anomaly score, was used to rank the data points from most to least anomalous..

One-class support vector machine: A one-class support vector machine (OC-SVM) is an outlier detection method that attempts to fit a boundary around normal data points and classifies new points as normal or abnormal based on their position relative to this boundary. The boundary can take the form of either a hyperplane or hypersphere, depending on the formulation of the problem. The OC-SVM model is trained using normal data and then classifies new data points based on their distance from the origin in the feature space.

One-class neural networks: One-class neural networks are designed to detect anomalies by training on a set of normal examples and then classifying new examples as normal or abnormal based on their deviation from the learned feature space of normal data. This method is particularly valuable in unsupervised learning scenarios, in which labelled data for the minority class are unavailable. It can be applied to binary classification tasks with highly imbalanced class distributions or in cases with no clear structure to differentiate classes using supervised algorithms..

3.3.5 Assessment criteria

A confusion matrix, or error matrix, summarises the predictions in matrix form for all classification algorithms considered. The matrix reveals direct values, such as true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Additional insights were gained through comparative analysis using metrics such as accuracy, precision, recall (sensitivity), specificity, G-mean, and F1-score.

For binary classes 1 and 0, TP and TN refer to the number of '1 s' that the model correctly classified as 1. In addition, the number of '0 s' was correctly classified as 0. Similarly, FP and FN gave the number of '1 s' that the model incorrectly classified as 0, and the number of '0 s' that were also incorrectly classified as 1. The other metrics are determined defined in Table 2.

Accuracy measures the proportion of correct predictions made by the model. However, in imbalanced datasets, the accuracy can be misleading, as the majority class may skew the results. Precision quantifies the proportion of true positive predictions among all the positive predictions made by the model, offering insight into its performance in the minority class. Recall (or sensitivity) evaluates the proportion of

Table 2 Evaluation metrics for imbalanced classification

Evaluation metrics	Formulas
Accuracy	$\frac{\text{True Positives} + \text{True Negatives}}{\text{Total Observations}}$
Precision	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
Recall	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$
Specificity	$\frac{\text{True negatives}}{\text{True negatives} + \text{False Positives}}$
G-Mean	$\sqrt{(\text{Sensitivity} \times \text{Specificity})}$
F1-score	$\frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$
AUC	$\frac{1 + \text{True Positive Rate} - \text{False Positive Rate}}{2}$

true positive predictions among all actual positive instances. Specificity measures the proportion of true negatives among all the actual negative instances. Additionally, the G-mean is the geometric mean of sensitivity and specificity, whereas the F1-score represents the harmonic mean of precision and recall, providing a balanced view of these metrics. While we acknowledge that MAUC offers a comprehensive evaluation of model performance in multiclass-imbalanced settings, it was not included in the present study as our primary focus was on binary and three-class classification tasks where other metrics such as AUC, G-mean, and F1-score are more commonly adopted and sufficient for performance comparison [45]. Higher values of these performance metrics indicated a better model.

4 Results and discussion

The discussion of the experimental results starts with the performance of the base classifiers on the synthetic and benchmark datasets without resampling across the three levels of imbalances. This is followed by the effects of oversampling and undersampling techniques on the classifiers and levels of imbalance. The outcome of implementing ensemble, cost-sensitive, and one-class classification models across imbalances has also been reported. The F1-score was used to present the performance of the algorithms across various imbalances. Deductions from the results were used to formulate the rules for the recommended system.

4.1 Performance of the base classifiers without resampling across levels of imbalances

The performances of the decision tree (DT), linear regression (LR), support vector machines (SVM), and artificial neural networks (ANN) on the extreme, moderate, and mild datasets are summarised in Table 3 for the synthesised dataset, and Table 4 for the benchmark dataset.

Table 3 Performance of base classifiers on synthesised dataset

Models	Classes	DT	LR	SVM	NN
F1-score—extreme	0	0.99	0.99	1.00	1.00
	1	0.04	0.00	0.92	0.00
F1-score—moderate	0	0.95	0.95	1.00	1.00
	1	0.25	0.24	0.92	0.90
F1-score—mild	0	0.85	0.85	1.00	1.00
	1	0.57	0.53	0.92	0.90

Table 4 Performance of base classifiers on benchmarked dataset

Models	Classes	DT	LR	SVM
F1-score—extreme	0	1.0	1.0	1.0
	1	0.82	0.7	0.8
F1-score—moderate	0	0.97	0.97	0.97
	1	0.0	0.0	0.0
F1-SCORE—MILD	0	0.86	0.88	0.88
	1	0.5	0.63	0.62

The performances of the base classifiers on the extremely imbalanced synthetic dataset for the majority and minority (0,1) classes show that the LR and NN perform extremely poorly, with (0.99,0.00) and (1.00, 0.00) for classes 0 and 1, respectively. The performance of DT is also insignificant with only 0.04 recognition of the minority group, whereas SVM performs very well in the minority class (1.00, 0.92) without losing sight of the majority class.

There was a marginal improvement in the performance of DT and LR on moderate imbalances of (0.95, 0.95) and (0.25, 0.24), respectively. The NN shows a significant improvement (1.00, 0.90) almost in per with the performance of SVM which is (1.00, 0.92). These performances are sustained on a mildly imbalanced dataset by the NN and SVM. DT and LR also have an improved performance of (0.85, 0.85) and (0.57, 0.53), respectively.

The base algorithms demonstrated various degrees of performance on the three benchmark datasets. As shown in Table 4, DT led on the extreme dataset with (1.0,0.82) followed by SVM (1.0,0.8) and LR (1.0, 0.7). All models have the same performance (0.97,0.0) for the moderate benchmark dataset. SVM has (0.88, 0.62) on the mild dataset to be in the second position, LR led with (0.88, 0.63), and the least performance from DT is (0.86,0.5).

These results reveal the tolerance levels of the four base classifiers to imbalanced datasets. This shows that the SVM is the most appropriate classifier for all forms of imbalanced datasets because of its lead performance and consistent second position on the benchmark datasets. The performance of the NN follows on the synthesised dataset, although the performance of the NN cannot be reported on a benchmark dataset for logistic reasons. DT is a better classifier for the

imbalanced dataset than LR, as demonstrated in the results from both the synthesised and benchmark datasets.

To assess whether the classifiers exhibit statistically detectable differences in their performances based on the observed F1-scores, we performed the Friedman test [46] to test the null hypothesis that the classifiers are equivalent with respect to the observed F1-scores, which yielded a significant result (p -value = 0.01775). This implies that there is a statistically significant difference in performance among DT, LR, and SVM. This further justifies the analytic inferences presented from the results.

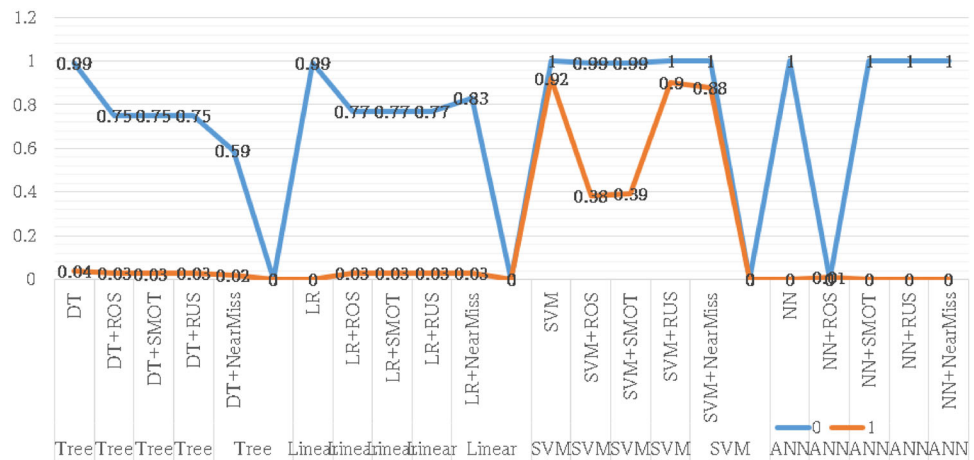
4.2 Implication of resampling on the performances of the classifiers across levels of imbalances

The two oversampling techniques used were ROS and SMOTE. Similarly, RUS and NearMiss were applied as the undersampling techniques. These four resampling methods were individually implemented on the four base models across the three levels of class imbalance for the two dataset categories.

4.2.1 Effects of resampling on the extremely imbalance dataset

Figure 2 shows the negative effects of resampling the extremely synthesised imbalanced dataset on the performance of the DT and SVM classifiers. The performance of the base DT for the majority (0) and minority (1) classes is (0.99, 0.04), and none of the resampling techniques met that performance; the least performance of (0.59, 0.02) was achieved by DT + NearMiss. The performance of SVM was

Fig. 2 Effect of rebalancing techniques on an extremely imbalanced synthetic dataset



also reduced from (1.00, 0.92) to (0.99, 0.38) by SVM + ROS. The best result from resampling is (1.00, 0.9) from SVM + RUS. A slight improvement on the minority recorded in the resampling of LR from (0.99, 0.0) to (0.77, 0.03) resulted in a reduced performance in the majority class.

Lower resampling performance is also observed in the extremely imbalanced benchmark datasets, as shown in Fig. 3. The DT performance on the credit card dataset is (1.0, 0.82). This value dropped to (1.0, 1.35) in DT + NearMiss, and further decreased to (0.97, 0.06) in DT + ROS.

The same trend was maintained for LR and SVM. LR has (1.0, 0.7), which was reduced to (0.99, 0.12) by LR + ROS as the highest and (0.95, 0.03) from LR + NearMiss as the lowest performance. SVM has (1.0, 0.8) performance, but when resampled with SMOT, the lowest value of (0.99, 0.11) was recorded, and the highest from the resampling techniques was (1.0, 0.77) from RUS.

4.2.2 Effect of resampling on the moderately imbalance dataset

The resampling of a moderately imbalanced dataset reduces the performance of the SVM and NN; however, it provides a slight improvement in the performance of DT and LR on the minority class, as shown in Fig. 4. However, this improvement on the minority in the two classifiers for the moderately imbalanced dataset resulted in low recognition of the majority class. The DT performance on the moderately imbalanced dataset is (0.95, 0.25), while the performances of DT + ROS is (0.75, 0.32), DT + SMOT is (0.77, 0.31), DT + ROS is (0.77, 0.32), and DT + NearMiss is (0.84, 0.33).

The same trend is maintained by LR with base value (0.95, 0.24), and the highest value on the minority from resampling is (0.85, 0.34) from LR + NearMiss the three other resample techniques generate a lower value of (0.8, 0.32). The performance of the SVM for the moderate imbalance dataset was (1.00, 0.92), which was reduced to (0.99, 0.38) by SVM

+ ROS as the lowest and (1.00, 0.88) by SVM + NearMiss as the highest from resampling. Similarly, the base NN has (1.00, 0.99), the highest from the resampling is the NN + ROS value of (0.99, 0.39), and the lowest is the (0.99, 0.36) from NN + RUS and NN + NearMiss.

A moderate improvement was observed in the minority class when resampling techniques were applied to the stroke dataset for all models. As shown in Fig. 5, the improvements range from 0.2 and 0.3.

Conversely, resampling lowers the performance of the models in the majority class from 0.97 through various values, to 0.81. All the base models performed poorly on the stroke dataset, and resampling provided marginal improvement across all the models.

4.2.3 Effect of resampling on the mild imbalance dataset

The results in Fig. 6 reveal that resampling mildly imbalanced datasets does not improve the performance of SVM and NN; instead, it reduces the performance across all rebalancing techniques. As shown, the base SVM performance on the mild dataset is (1.0, 0.92); this value reduces to (0.99, 0.38) by SVM + ROS as the lowest and (1.0, 0.9) as the highest from resampling. Similarly, the NN performance value was (1.0, 0.9), and all the values from the resample were lower, with NN + RUS and NN + NearMiss values of (0.99, 0.38) being the lowest.

The base DT performance on mild dataset is (0.85, 0.57); DT + NearMiss ups this performance with 0.86 for the majority and equals the performance with 0.57 for the minority classes. The performance of ROS, SMOT, and RUS for the minority class was 0.6, while the performance for the majority class for ROS and RUS was 0.76, and that of SMOT was 0.74. The LR value (0.85, 0.53), the LR + NearMiss resampling produce (0.81, 0.59), all other rebalances recorded a lower value of (0.77, 0.59).

Fig. 3 Effect of rebalancing techniques on the extremely imbalanced credit card dataset

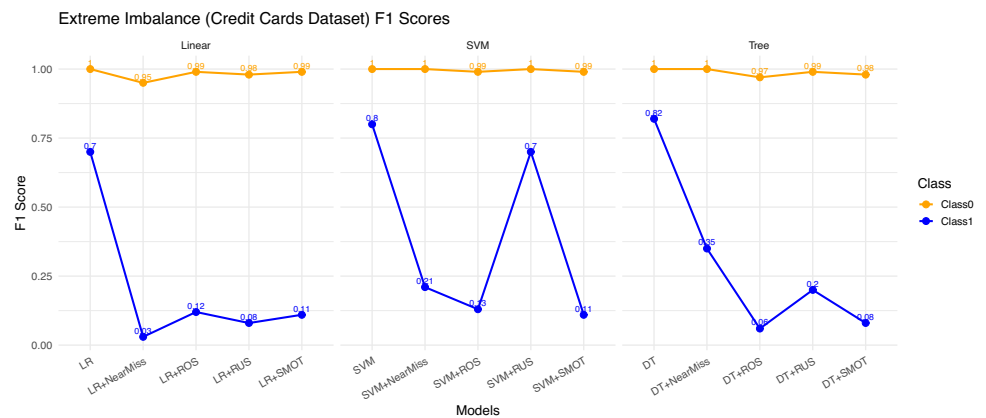


Fig. 4 Effect of rebalancing techniques on the moderate imbalanced synthetic dataset

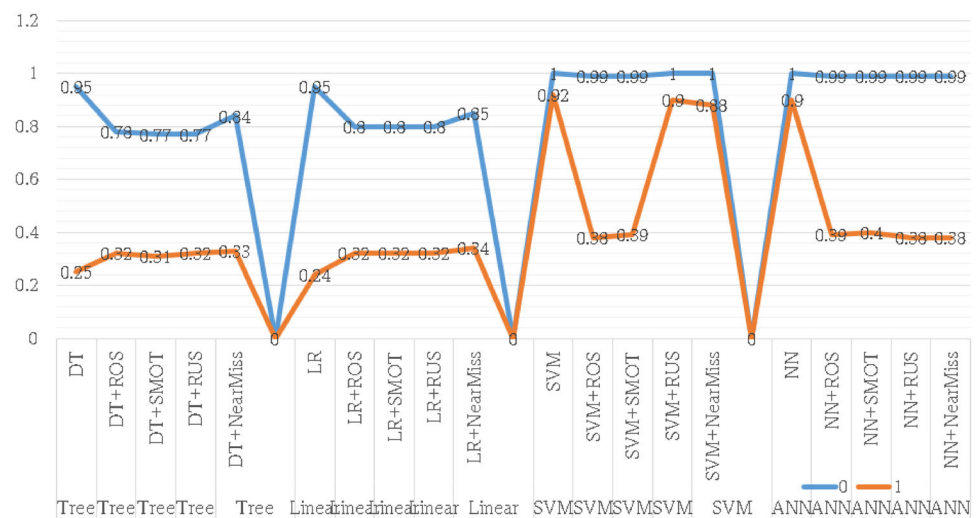
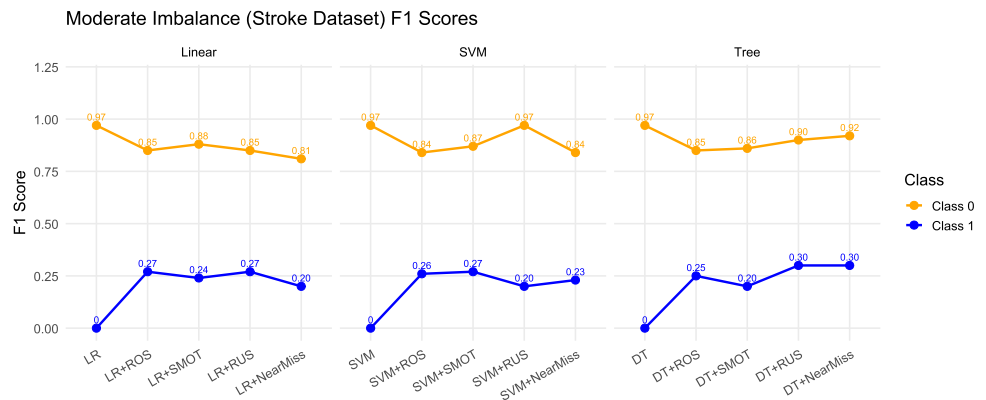


Fig. 5 Effect of rebalancing techniques on the moderately imbalanced stroke dataset



The result from the customer churn dataset shown in Fig. 7 revalidates the performance recorded with the mild synthetic imbalance datasets. In the minority class, the resampling techniques improved by 0.5 of the base DT, up to 0.61. In addition, resampling marginally improves the performance of LR on the minority from 0.63 to 0.64. The same value of 0.62 is recorded by the base SVM and the resample dataset with ROS, SMOT, and NearMiss. For the minority class, SVM + RUS has 0.63 for a minority class.

For all instances except SVM + RUS, the resampling techniques reduce the performance of the model on the majority class for the mildly imbalanced dataset.

The effect of rebalancing was low on the SVM and NN across the extreme, moderate, and mild imbalanced datasets. These results conform to those of [15] in which MLPs and SVMs were less affected by resampling. Also Thanka, Ram [23] claimed that ANN models performed satisfactorily on

Fig. 6 Effect of rebalancing techniques on a mildly imbalanced synthetic dataset

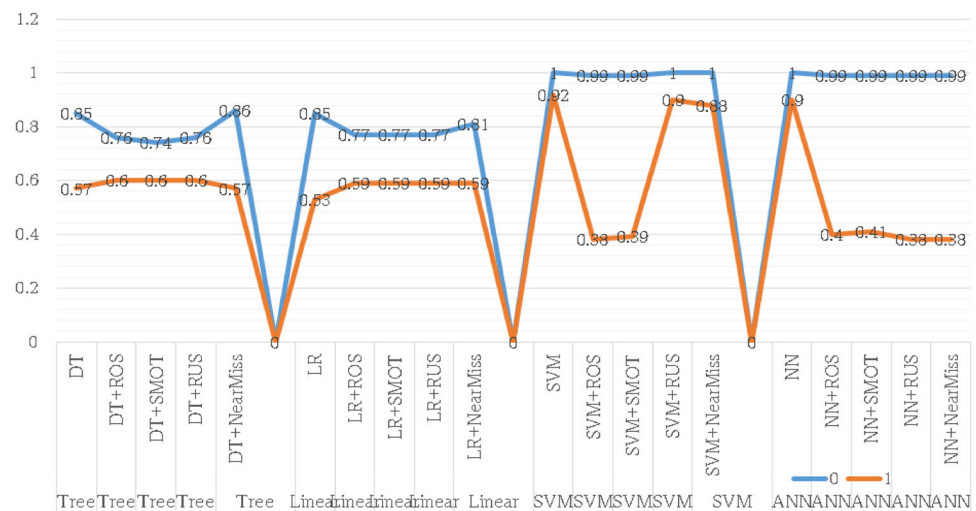
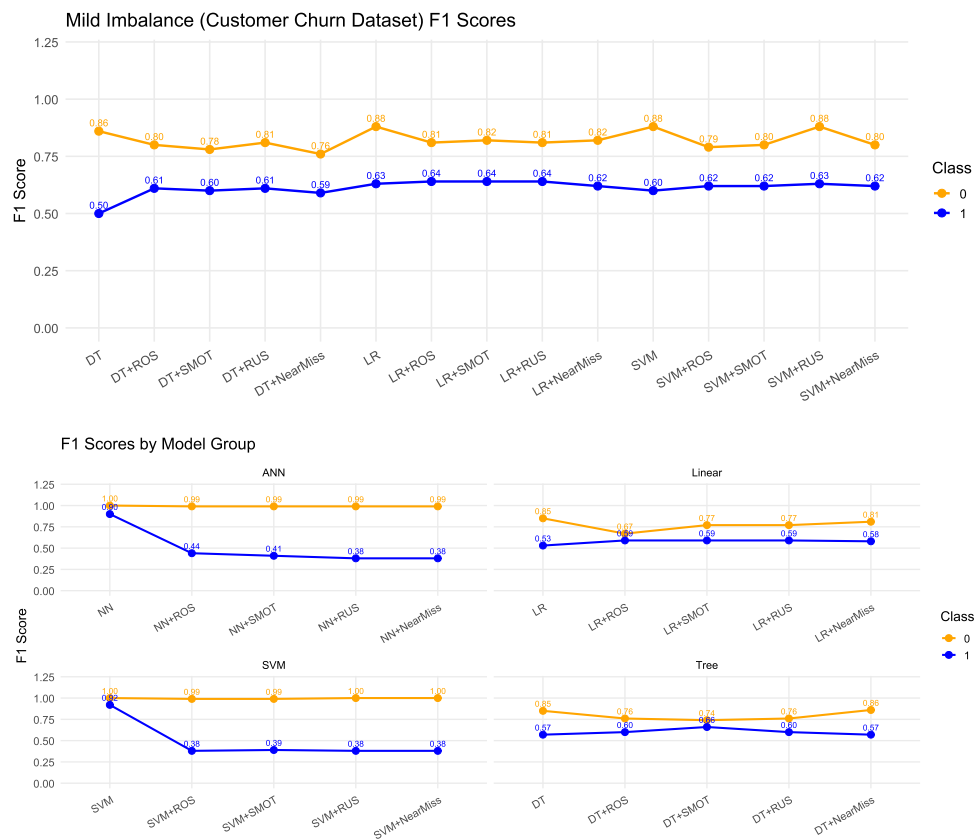


Fig. 7 Effect of rebalancing techniques on the mildly imbalanced customer churn dataset



the original imbalanced dataset and resampled dataset. Furthermore, this result justifies the fact that SVM and NN are the most appropriate for direct use on all forms of imbalanced datasets.

Furthermore, the effect of resampling is not sufficiently high to significantly influence a non-performing classifier to become an excellent classifier. The best approach is to sustain the performances as seen in RUS and NearMiss on SVM. There are also instances in which resampling reduces the

performance of the models, as seen in the NN. This outcome agrees with that in [4] that sampling has little or no effect on the performance of most classifiers. The next section reports the performance of the ensemble, cost-sensitive, and one-class classification on the imbalance classification.

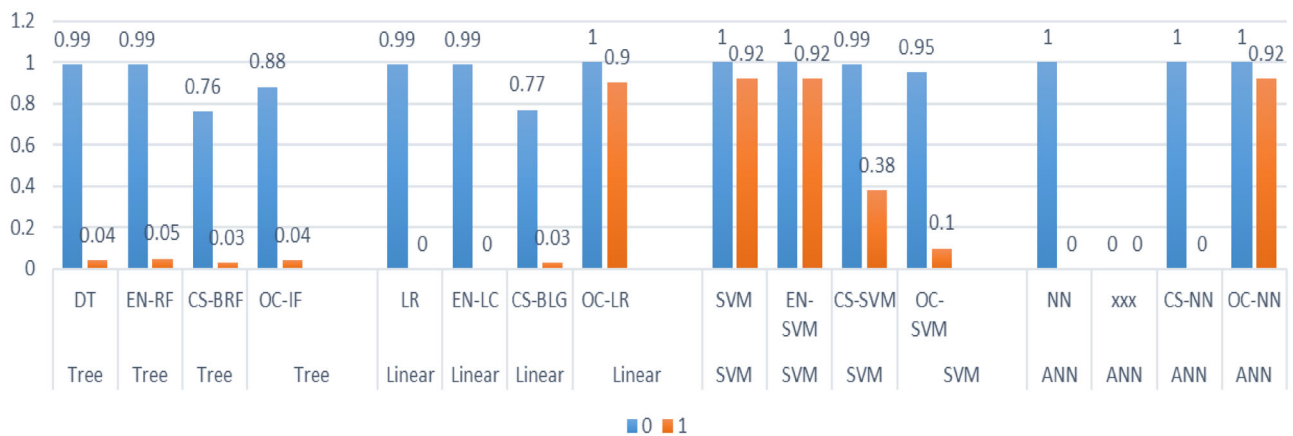


Fig. 8 Ensemble, cost-sensitive, and one-class classification performance on extreme imbalance synthetic dataset

4.3 Implication of ensemble, cost-sensitive and one-class classification on imbalance classification

The performance of the ensemble of binary tree—random forest (EN-RF) on the extremely imbalanced synthetic dataset as shown in Fig. 8 is (0.99, 0.05), that of ensemble linear regression (EN-LR) is (0.99, 0.00), the value for the ensemble SVM (EN-SVM) is (1.00, 0.92). That of ensemble ANN is (0.00, 0.00). Except for EN-RF, which outperforms the base DT by 0.01, no other ensemble models outperform their base models on the extremely imbalanced dataset. The values of the cost-sensitive models for the extremely imbalanced dataset were not up to the values of their base models. The performance of OC-IF (0.88, 0.04) is in accordance with that of the base DT (0.99, 0.04) on the minority class, and OC-LR and OC-NN performed much better than their base LR and NN, respectively. Therefore, the one-class model stands out as better for an extremely imbalanced dataset. The EN-RF and EN-SVM were good models for this category.

The results from the credit cards dataset in Fig. 9 further corroborate the outcomes of the synthetic dataset.

As shown, EN-RF and OC-LR outperform and perform the base models.

The F1-scores of models on the moderate imbalance synthetic dataset in Fig. 10 show that the performance of ensemble models is equal to that of the base models or moderately better than that of the base models. The former is as seen in LR (0.95, 0.24)/EN-LR (0.95, 0.24) and SVM (1.0, 0.92)/EN-SVM (1.0, 0.92). The latter is as seen in DT (0.95, 0.25) and EN-RF (0.85, 0.36). CS-BRF (0.81, 0.32) and CS-BLG (0.8, 0.32) outperformed the corresponding base models. CS-SVM (0.99, 0.38) and CS-NN (0.00, 0.01) performed lower than their base models. The performance of the cost-sensitive models is better for moderately imbalanced datasets, followed by the one-class model techniques.

The deduction from the synthetic moderately imbalanced dataset is substantiated by the output from the stroke dataset, as shown in Fig. 11.

CS-BRF, CS-BLG, and CS-SVM outperformed their corresponding base models and all other types of models for the moderately imbalanced real dataset demonstrated using the stroke dataset.

For the mildly imbalanced dataset, Fig. 12 shows the results for the synthetic dataset, OC-IF (0.78, 0.37), and OC-SVM (0.95, 0.10) performed below their base models. On the other hand, CS-BRF (0.78, 0.59) and OC-LR (1.00, 0.90) outperformed their base models, whereas the performance of OC-NN (1.00, 0.90) was equal to that of the base model.

The performance of the synthetic mild imbalance dataset is reflected in the output of the customer churn dataset in Fig. 13. The result of CS-BRF (0.81, 0.63) is better in the minority class than the base model DT (0.86, 0.5). EN-LC and OC-LR have the same performance (0.88, 0.63) as the base model LR. The OC-BLG (0.81, 0.63) is the same as the others in the minority class, but lower than the others in the majority group.

Comparatively, there was little or no improvement in the performances of EN-RF, EN-LC, and EN-SVM over the performances of DT, LR, and SVM, respectively, across the three levels of imbalances. The study could not experiment with the EN-NN because of the huge space and time requirements. Most studies on ensembles provide information on performance without reporting the significance of the performance over the base models.

In the same vein, a modest improvement was displayed by CS-BRF on moderate and mild imbalances, but no improvement was noticed on extremely imbalanced datasets. Unlike CS-BRF, CS-BLG performed better across all three levels of imbalances. The performances of CS-SVM and CS-NN were extremely low across the three imbalance levels.

Fig. 9 Ensemble, cost-sensitive, and one-class classification performance on credit cards dataset

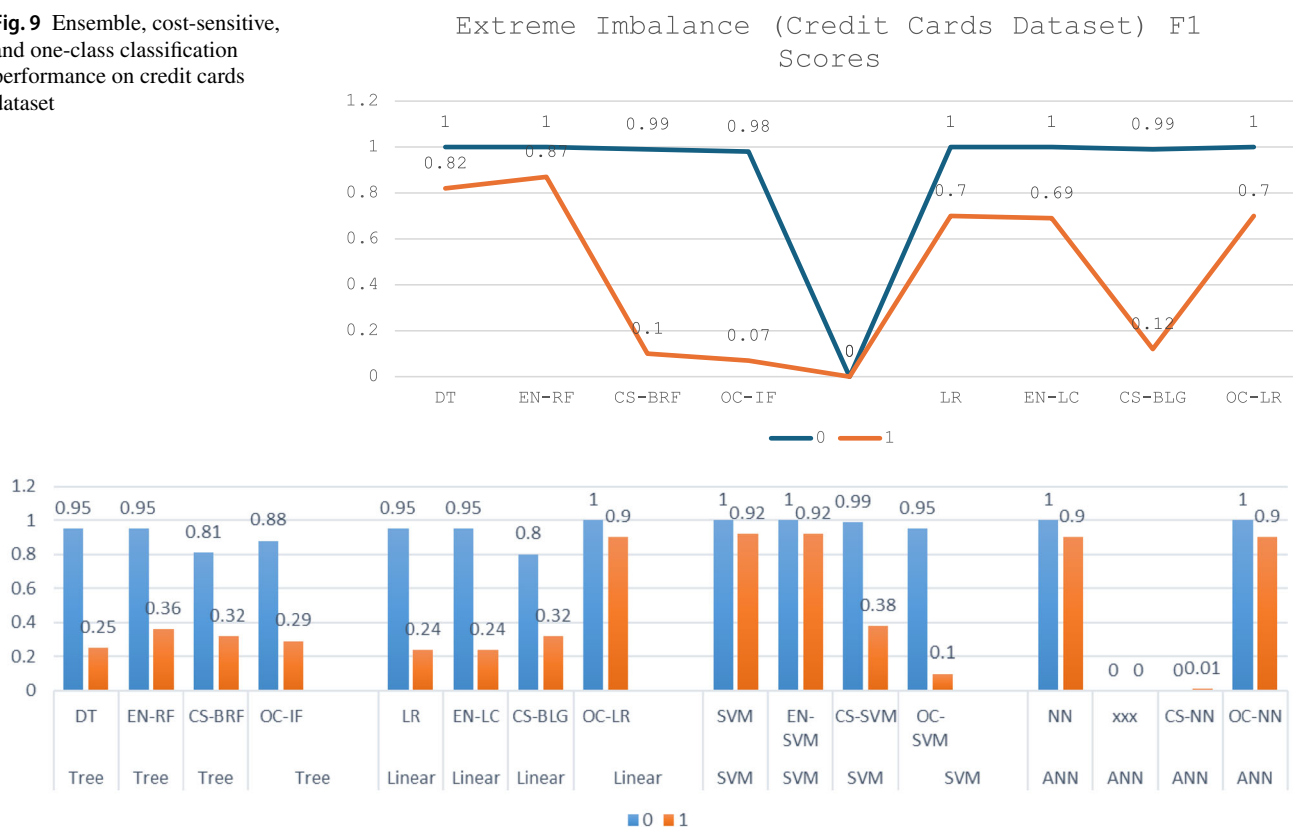


Fig. 10 Ensemble, cost-sensitive, and one-class classification performance on moderate imbalance synthetic dataset

Fig. 11 Ensemble, cost-sensitive, and one-class classification performance on stroke dataset

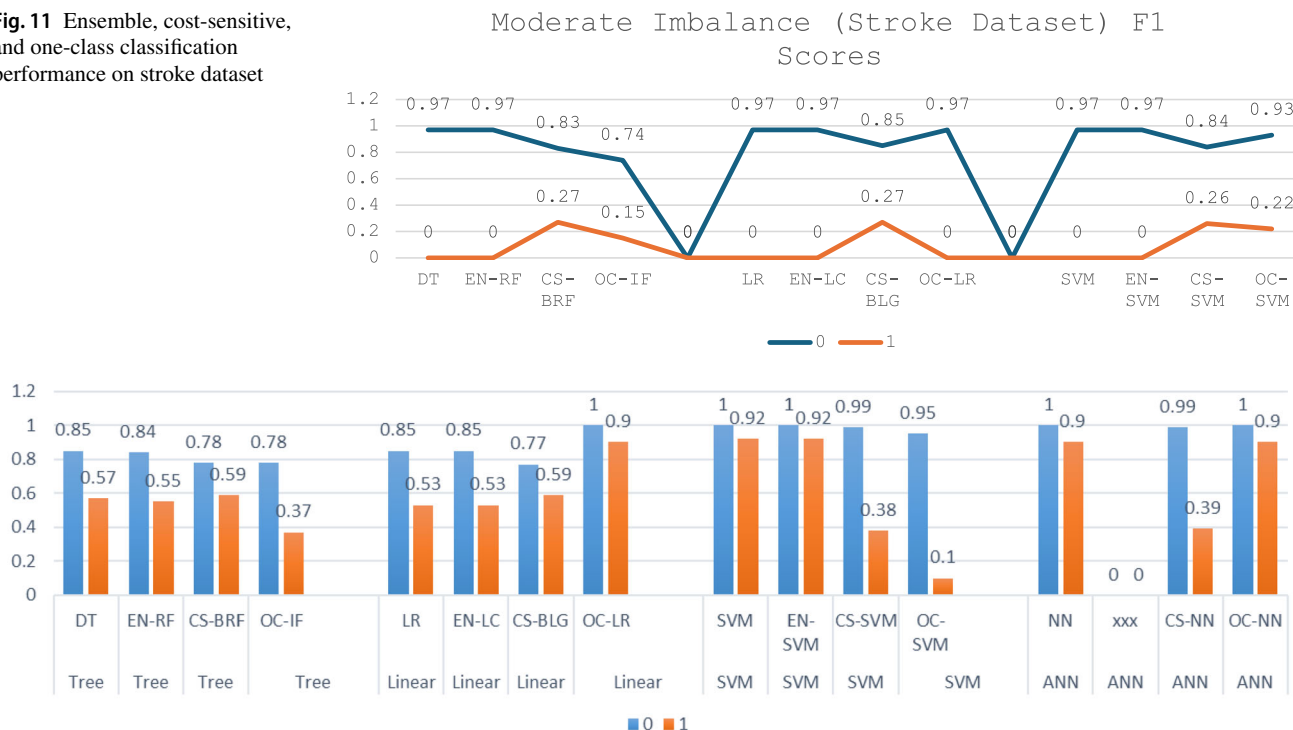
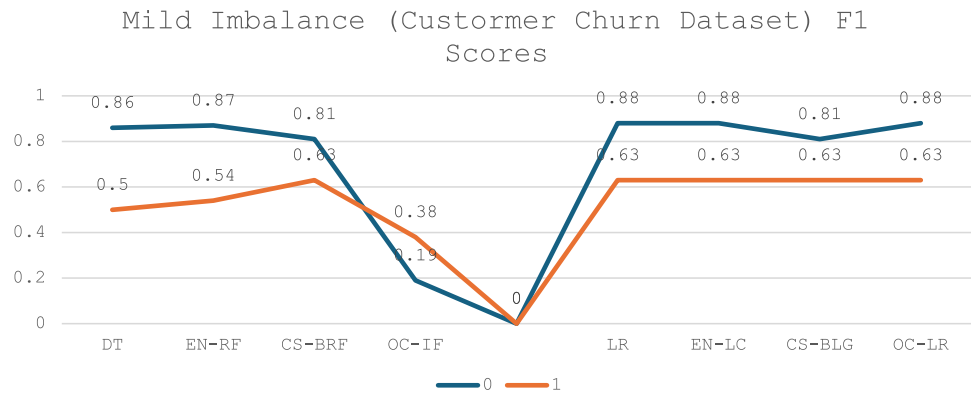


Fig. 12 Ensemble, cost-sensitive, and one-class classification performance on mild imbalance synthetic dataset

Fig. 13 Ensemble, cost-sensitive, and one-class classification performance on customer churn dataset



However, the performance of one-class classification techniques is significant across all levels of imbalances for OC-LR, OC-SVM, and OC-NN, and the improved performance in OC-IF is not as high as that of the other three models.

5 Modelling of a knowledge-based recommender system for imbalance classification

The performance of the resampling techniques, ensembles, cost sensitivity, and one-class classification on the four classifiers and across the three levels of imbalances is coded, as shown in Table 5. The value chosen as the threshold of average performance on the minority class is 0.5 (f), values below the threshold are considered unsatisfactory 0 (X), and values above the threshold are rated satisfactory 1 ($\checkmark$).

In Table 5, the overall performance of DT is 2 that of LR is 5.5, that of SVM is 15, and that of NN is 5. These results indicate that the SVM is the most effective algorithm for the direct handling of class-imbalanced datasets, followed by LR and NN. DT is the least effective algorithm for managing class imbalances in binary classification problems.

Similarly, the results reveal that undersampling is a better resampling technique than oversampling with $RUS = 3.5/12$ and $NearMiss = 4/12$ while $ROS = 0.5/12$ and $SMOT = 0.5/12$.

Although the performance of EN-NN is not reported here because of its computational complexity, the overall performance in four out of nine cases for the other three scenarios indicated that ensembles outperformed both undersampling and oversampling techniques. The one-class classification technique achieved a performance of 8 out of 12, making it the best overall model for handling imbalanced learning problems. This was followed by the ensemble, undersampling, and cost-sensitive techniques. Oversampling techniques performed the worst among all methods considered.

5.1 Formulation of imbalance learning recommender system's model

The twenty-seven (27) rules derived from Table 5 are encoded in Table 6 and used to develop the imbalanced recommender system model, as illustrated in Figs. 14, 15, and 16. These rules are expressed as a series of IF-THEN statements that capture the explicit domain knowledge.

The selection of a knowledge-based recommender system is driven by the transparent and explainable nature of the rules or knowledge representations used by these techniques.

When a dataset is provided, the system determines the level of imbalance and specifies the recommended options for the user. However, the actual choice that is eventually made by the user will depend on other business rules and necessary conditions that must be considered. A typical output for a mildly imbalanced dataset is shown in Fig. 14.

Figure 14 shows the recommendations for the mild dataset. This indicates that DT, SVM, and NN can be used directly on the dataset. If resampling is required, *RUS* and *NearMiss* are preferable. EN-RF and EN-SVM were the ensemble models recommended in this case. The recommended cost-sensitive model was CS-BRF, and the one-class models were OC-LR, OC-SVM, and OC-NN.

The default recommendations for the moderate and extreme datasets are shown in Fig. 15 and 16, respectively.

The models are the actualisation of the recommendation engine and the user guidance component in the *imbalanced learning recommender framework* shown in Fig. 1. The model's output Figs. 14, 15 and 16 displays the default actual performance values for each data level as a guide for the user. The user's actions and transaction outcomes will serve as feedback for the system towards continuous improvement.

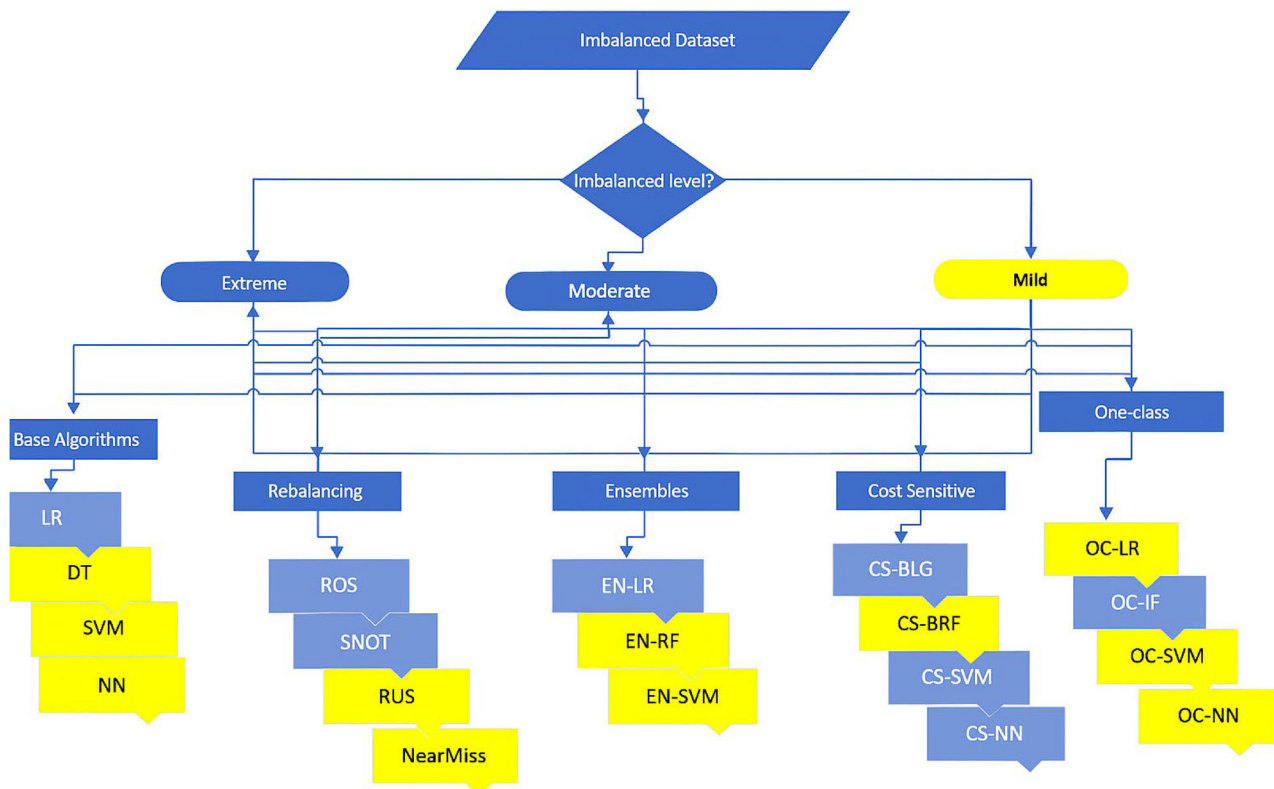


Fig. 14 Model recommendation for mild imbalance data

6 Summary, conclusion, and recommendations

Properly managing class-imbalanced learning is essential in machine learning, data mining, and other data-driven problem-solving approaches. The resampling techniques that are commonly used to address this problem normalise the dataset classes to equal 50% either by oversampling with artificially generated data or undersampling by losing some original information in the dataset. Therefore, models built on resampled datasets are based on artificial assumptions, with inconsistent accuracy and unsatisfactory performance when deployed in the real-world domain. Therefore, it is essential to demonstrate the capacity of other approaches to handle imbalanced learning without the side effects associated with resampling methods.

It is evident from this study that resampling is not necessary in imbalanced learning, because it is not always the best solution. The performance of the resampled models ranked them on par with or below, but certainly not above, other available alternative techniques, such as ensemble methods, cost-sensitive learning, and one-class classification techniques.

Hence, with the limitations attributed to rebalancing the class-imbalanced dataset, it is essential for the imbalanced learning research community and users not to be carried away by the hype on the performance of the resampling techniques; instead, safer algorithms and hybrid approaches should be explored to handle class-imbalanced problems. Furthermore, a recommender system is provided in this study to provide the necessary guidance for users to save time and effort expended on the search for the best techniques for a given problem.

Future work on this study would consider the exploration of more classifiers and datasets to further demonstrate the adequacy of the findings with an extension to multiclass-imbalance datasets. In addition, the scalability of one-class logistic regression and other techniques on datasets with high dimensionality or significant noise was explored. Furthermore, the recommender system can be deployed and made accessible to an unbalanced learning community. The usability evaluation and other feedback received can be used for the continuous enhancement of the platform.

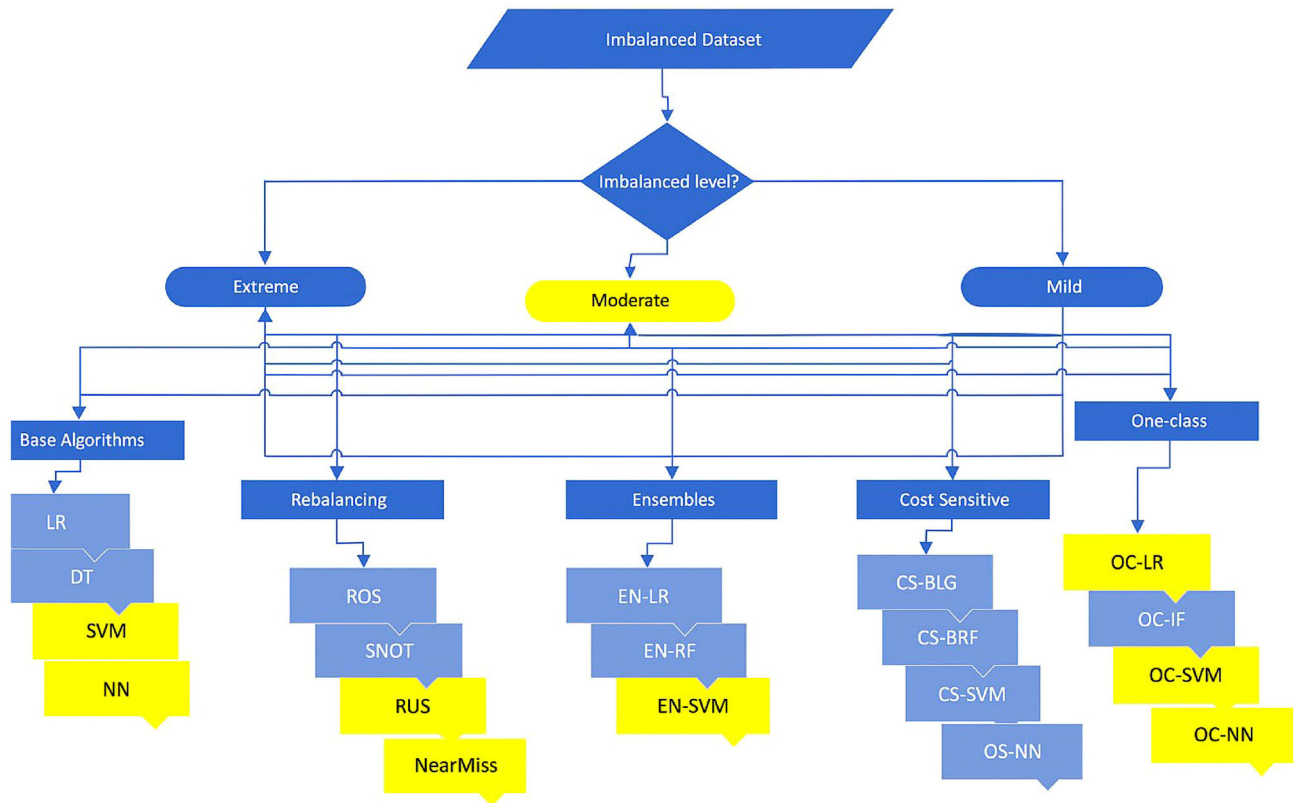


Fig. 15 Model recommendation for moderate imbalance data

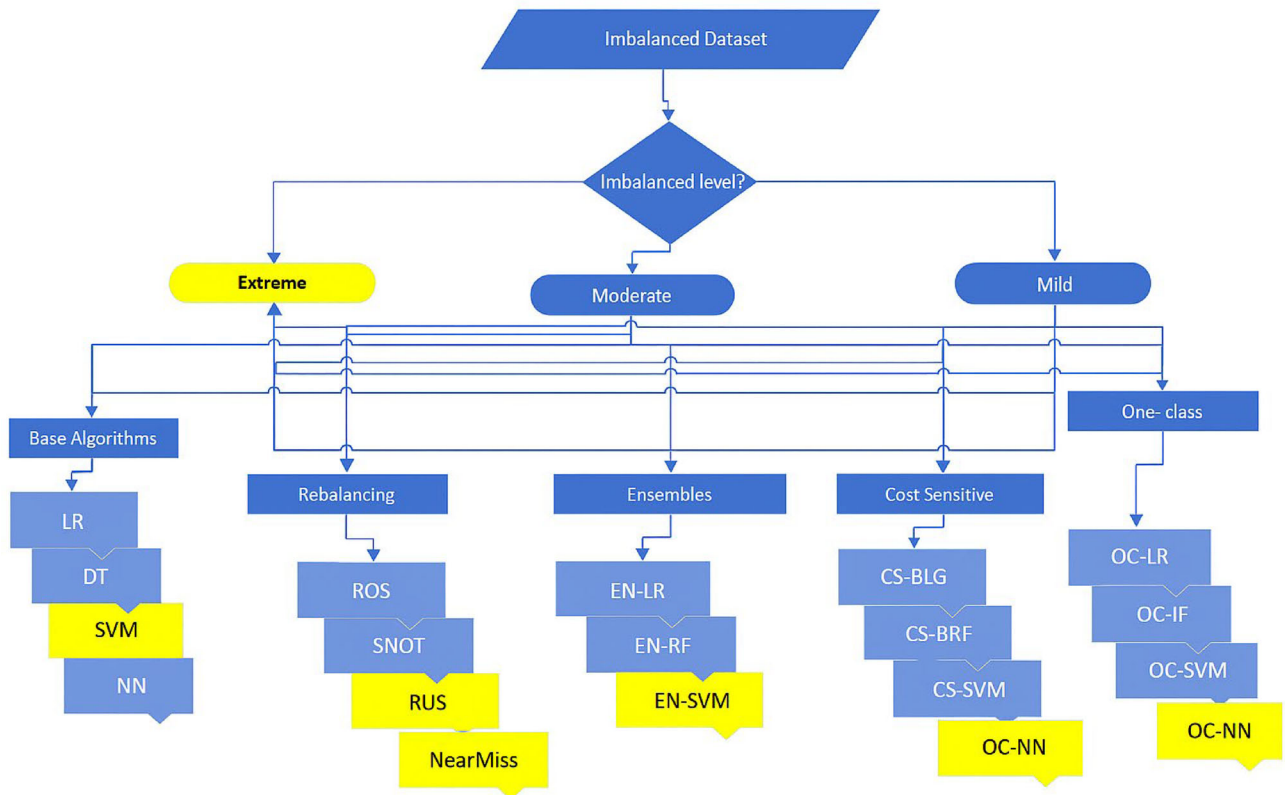


Fig. 16 Model recommendation for extreme imbalance data

Table 5 Performance of the techniques over the three levels of imbalances

Models	Classes	DT	DT + ROS	DT + SMOT	DT + RUS	DT + NearMiss	EN-RF	CS-BRF	OC-IF	
F1-score—extreme	1	X	X	X	X	X	X	X	X	0
F1-score—moderate	1	X	X	X	X	X	X	X	X	0
F1-score—Mild	1	f	X	X	X	f	f	f	X	2
Decision tree (DT)										2
F1-score—extreme	1	LR	LR + ROS	LR + SMOT	LR + RUS	LR + NearMiss	EN-LC	CS-BLG	OC-LR	0
F1-score—moderate	1	X	X	X	X	X	X	X	X	1
F1-score—mild	1	f	f	f	f	f	f	f	f	4.5
Linear regression (LR)										5.5
F1-score—extreme	1	SVM	SVM + ROS	SVM + SMOT	SVM + RUS	SVM + NearMiss	EN-SVM	CS-SVM	OC-SVM	5
F1-score—moderate	1	✓	X	X	✓	✓	✓	X	✓	5
F1-score—mild	1	✓	X	X	✓	✓	✓	X	✓	5
Support vector machine (SVM)										15
F1-score—extreme	1	NN	NN + ROS	NN + SMOT	NN + RUS	NN + NearMiss	EN-NN	CS-NN	OC-NN	1
F1-score—moderate	1	X	X	X	X	X	-	✓	✓	2
F1-score—mild	1	✓	X	X	X	X	-	X	✓	2
Neural network (NN)										5
Overall model performance	6/12	0.5/12	0.5/12	3.5/12	4/12	4/9	2/12	8/12		

Table 6 Premises and the consequents of the twenty-seven (27) rules

	< Premises >										> Consequents >									
	SVM	SVM + RUS	SVM + NearMiss	EN-SVM	OC-SVM	OC-NN	CS-NN	OC-LR	NN	DT	DT + NearMiss	EN-RF	CS-BRF							
IF data imbalanced level = extreme THEN	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-							
IF data imbalanced level = moderate THEN	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-							
IF data imbalanced level = mild THEN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓							

Author contribution RMI: Conceptualisation, data curation, formal analysis, investigation, methodology, resources, software, validation, visualisation, Writing the original draft, supervision, manuscript revision KOO: Conceptualisation and co-ideation, data curation, methodology correction and update, resources, software, visualisation, Writing review and editing, submission, manuscript revision RSB: Co-ideation, investigation, review, project administration, resources FJA: Co-ideation, investigation, review, project administration, resources.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interests The authors declare no conflict of interests.

References

- Hassanat, A.B., et al.: Stop oversampling for class imbalance learning: A critical review. arXiv preprint [arXiv:2202.03579](https://arxiv.org/abs/2202.03579) (2022)
- He, H., Ma Y.: Imbalanced learning: foundations, algorithms, and applications (2013)
- Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **18**(17), 1–5 (2017)
- Kim, M., Hwang, K.-B.: An empirical evaluation of sampling methods for the classification of imbalanced data. *PLoS ONE* **17**(7), e0271260 (2022)
- Błaszczczyk, M., Jędrzejowicz, J.: Framework for imbalanced data classification. *Procedia Comput. Sci.* **192**, 3477–3486 (2021)
- Mulugeta, G., et al.: Classification of imbalanced data using machine learning algorithms to predict the risk of renal graft failures in Ethiopia. *BMC Med. Inform. Decis. Mak.* **23**(1), 98 (2023)
- Provost, F.: Machine learning from imbalanced data sets 101. In: *Proceedings of the AAAI'2000 workshop on imbalanced data sets*. AAAI Press (2000)
- Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.* **5**(4), 221–232 (2016)
- Haixiang, G., et al.: Learning from class-imbalanced data: review of methods and applications. *Expert Syst. Appl.* **73**, 220–239 (2017)
- Obaid, W., Nassif A.B.: The effects of resampling on classifying imbalanced datasets. In: *2022 Advances in Science and Engineering Technology International Conferences (ASET)*. IEEE (2022)
- Pujianto, U., et al.: The effect of resampling on classifier performance: an empirical study. *Knowl. Eng. Data Sci.* **5**(1), 87–100 (2022)
- de Zarzà, I., de Curtò, J., Calafate, C.T.: Optimizing neural networks for imbalanced data. *Electronics* **12**(12), 2674 (2023)
- Cicak, S., Avcı U.: Handling Imbalanced Data in Predictive Maintenance: A Resampling-Based Approach. In: *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE (2023)
- Nguyen, S., et al.: Effects of resampling techniques on imbalanced data classification: A new under-resampling method, in *Advances in Business and Management Forecasting*. Emerald Publishing Limited, p. 51–70 (2021)
- Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. *Intell. Data Anal.* **6**(5), 429–449 (2002)
- Zhao, Y., Wong, Z.S.-Y., Tsui, K.L.: A framework of rebalancing imbalanced healthcare data for rare events' classification: a case of look-alike sound-alike mix-up incident detection. *J. Healthcare Eng.* **2018**(1), 6275435 (2018)
- Zhang, J., et al.: A class-aware supervised contrastive learning framework for imbalanced fault diagnosis. *Knowl.-Based Syst.* **252**, 109437 (2022)
- Jiang, E.P.: A hybrid learning framework for imbalanced classification. *Int. J. Intell. Inf. Technol.* **18**(1), 1–15 (2022)
- Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. *J. Big Data* **6**(1), 1–54 (2019)
- Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *J. Big Data* **6**(1), 1–48 (2019)
- van den Goorbergh, R., et al.: The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression. *J. Am. Med. Inform. Assoc.* **29**(9), 1525–1534 (2022)
- Harrell, F.: Classification versus prediction. *Statistical Thinking* (2020)
- Thanka, M.R., et al.: Comparing Resampling Techniques in Stroke Prediction with Machine and Deep Learning. In: *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. IEEE (2023)
- Tarawneh, A.S., et al.: Stop oversampling for class imbalance learning: a review. *IEEE Access* **10**, 47643–47660 (2022)
- Liu, N., et al.: A novel ensemble learning paradigm for medical diagnosis with imbalanced data. *IEEE Access* **8**, 171263–171280 (2020)
- Shah, M., et al.: Theoretical evaluation of ensemble machine learning techniques. In: *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE (2023)
- An, X., Xu, S.: A selective evolutionary heterogeneous ensemble algorithm for classifying imbalanced data. *Electron. Res. Arch.* **31**(5), 2733–2757 (2023)
- Zhu, J., Zhang, A., Zheng, H.: Research on predictive model based on ensemble learning. *Highlights Sci. Eng. Technol.* **57**, 311–319 (2023)
- Wang, C., et al.: Adaptive ensemble of classifiers with regularization for imbalanced data classification. *Inf. Fusion* **69**, 81–102 (2021)
- Wang, N., et al.: Cost-sensitive hypergraph learning with F-measure optimization. *IEEE Trans. Cybern.* **53**(5), 2767–2778 (2023)
- Awasthi, M., et al.: Performance efficacy of cost-sensitive artificial neural network: augmenting the results of imbalanced datasets in supervised and unsupervised learning. In: *Proceedings of International Conference on Communication and Computational Technologies: ICCCT 2022*. Springer (2022)
- Mienye, I.D., Sun, Y.: Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Inf. Med. Unlocked* **25**, 100690 (2021)
- Hayashi, T., Fujita, H.: One-class ensemble classifier for data imbalance problems. *Appl. Intell.* **52**(15), 17073–17089 (2022)
- Chen, W., et al.: A survey on imbalanced learning: latest research, applications and future directions. *Artif. Intell. Rev.* **57**(6), 137 (2024)
- Kushwaha, A., Pandey, R.S.: Imbalanced dataset classification using fuzzy ARTMAP and computational intelligence techniques. *Indones. J. Electr. Eng. Comput. Sci.* **30**(2), 909–916 (2023)
- Jafarigol, E., Trafalis T.B.: Federated learning with GANs-based synthetic minority over-sampling technique for improving weather prediction from imbalanced data (2023)
- Wu, J., et al.: Learning from class-imbalanced data with a model-agnostic framework for machine intelligent diagnosis. *Reliab. Eng. Syst. Saf.* **216**, 107934 (2021)
- Hastie, T., Tibshirani, R., Wainwright, M.: Statistical learning with sparsity. *Monogr. Stat. Appl. Probab.* **143**(143), 8 (2015)
- Mun, J., Bang, S., Kim, J.: Weighted support vector machine for extremely imbalanced data. *Comput. Stat. Data Anal.* **203**, 108078 (2025)

40. Boyapati, M., Aygun, R.: BalancerGNN: balancer graph neural networks for imbalanced datasets: a case study on fraud detection. *Neural Netw.* **182**, 106926 (2025)
41. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
42. Goodfellow, I.: Deep learning. Vol. 196. MIT press (2016)
43. He, H., Garcia, E.: Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009)
44. Chugh, B., et al.: A probabilistic approach driven credit card anomaly detection with CBLOF and isolation forest models. *Alex. Eng. J.* **114**, 231–242 (2025)
45. Chen, W., et al.: A survey on imbalanced learning: latest research, applications and future directions. *Artif. Intell. Rev.* **57**(6), 1–51 (2024)
46. Pohlert, T.: PMCMRplus: calculate pairwise multiple comparisons of mean rank sums extended (2024)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.