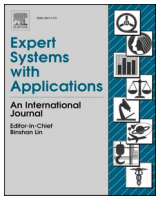
Expert Systems With Applications 262 (2025) 125674

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Mitigating false negatives in imbalanced datasets: An ensemble approach

Marcelo Vasconcelos [a,*], Luís Cavique [b]

[a] Tribunal de Contas do Distrito Federal, Brasília, Brazil
[b] Universidade Aberta and Lasige-FCUL, Lisboa, Portugal

ARTICLE INFO

Keywords:
Imbalanced dataset
False negative rate
Ensemble algorithms
Fraud detection
Set covering problem

ABSTRACT

Imbalanced datasets present a challenge in machine learning, especially in binary classification scenarios where one class significantly outweighs the other. This imbalance often leads to models favoring the majority class, resulting in inadequate predictions for the minority class, specifically in false negatives. In response to this issue, this work introduces the MinFNR ensemble algorithm, designed to minimize False Negative Rates (FNR) in imbalanced datasets. The new approach strategically combines data-level, algorithmic-level, and hybrid-level approaches to enhance overall predictive capabilities while minimizing computational resources using the Set Covering Problem (SCP) formulation. Through a comprehensive evaluation of diverse datasets, MinFNR consistently outperforms individual algorithms, showing its potential for applications where the cost of false negatives is substantial, such as fraud detection and medical diagnosis. This work also contributes to ongoing efforts to improve the reliability and effectiveness of machine learning algorithms in real imbalanced scenarios.

1. Introduction

Dealing with imbalanced datasets in binary classification is a challenging problem since the distribution of target attributes among classes is skewed. Imbalanced data is a common issue where models excel in predicting the majority class but struggle to identify instances from the minority class. Real-world applications such as fraud detection (Lebichot et al., 2021) and medical diagnosis (William et al., 2018) involve imbalanced problems.

In some imbalanced datasets, the impact of false negatives is more severe than false positives. For example, a false negative in intrusion detection represents a missed security breach, similar to a "perfect crime" going unnoticed. On the other hand, a false positive is like a "false alarm," indicating a breach where there is none. This distinction highlights the critical impact between a false negative and a false positive in intrusion detection. The cost of a false negative is said to be greater than a false positive's.

Knowles (Knowles et al., 2023) draw attention to a tendency in the artificial intelligence (AI) domain to underestimate the impact of false negatives. This oversight could have adverse consequences in the context of decision-making and risk assessment, as well as broader concerns related to the trustworthiness of AI systems.

For problems involving real cases of imbalanced datasets where the cost of false negatives is higher than that of false positives, the False Negative Rates (FNR) indicator becomes more relevant. Minimizing the FNR is crucial as it leads to improved predictions by the algorithms. This work introduces a novel ensemble algorithm, MinFNR, specifically designed to minimize the FNR.

Central to the effectiveness of the MinFNR algorithm is the Set Covering Problem (SCP), which is used to identify the optimal subset of classifiers that collectively minimize the FNR.

The SCP is a classic optimization problem that seeks to find the smallest subset of sets that covers all elements in a given universe. For MinFNR, the SCP selects the most relevant classifiers from a pool of candidates, ensuring that all positive instances are correctly identified while minimizing the number of classifiers used.

Boolean expressions represent the outcome of the SCP, indicating the selection of classifiers for a dataset. The conjunction of the composition of these expressions of the datasets results in the subset of classifiers that collectively minimize the FNR. By leveraging the SCP, MinFNR achieves superior performance in handling imbalanced datasets and offers a systematic approach to ensemble learning that can be applied to various applications related to imbalanced datasets.

In this paper, we present a detailed analysis of the MinFNR algorithm, demonstrating its effectiveness and discussing its implications for the field of machine learning. In Fig. 1, there is the flow process to create the MinFNR. It illustrates the steps to generate improved prediction results for highly imbalanced datasets. The imbalanced data from the

* Corresponding author.
E-mail addresses: mov@tc.df.gov.br (M. Vasconcelos), luis.cavique@uab.pt (L. Cavique).

https://doi.org/10.1016/j.eswa.2024.125674
Received 26 December 2023; Received in revised form 6 May 2024; Accepted 30 October 2024
Available online 3 November 2024

database was subjected to several algorithms and approaches designed to address data imbalance issues. Subsequently, the outcomes were consolidated through an optimization algorithm based on FNR to identify the most effective subset of algorithms. This refined subset was used to form the MinFNR, which represents the optimal combination for addressing the imbalances in the data.

1.1. Problem

The article addresses the challenge of obtaining the best prediction in binary classification tasks for actual cases of data imbalance in which the minority class is the most relevant and represents a high disproportion concerning the majority class. In these scenarios, conventional machine learning models tend to perform well in the majority class but struggle with the minority class, leading to high false negative rates.

1.2. Objective

The primary objective of this research is to develop a robust solution that effectively addresses the intricacies of imbalanced datasets in situations where the consequences of false negatives carry a higher cost than false positives, addressing real-world challenges in areas such as fraud detection, medical diagnosis, and similar domains. By focusing on minimizing FNR, we aim to enhance the predictive capabilities of machine learning models.

1.3. Contributions

The contribution of this article is the introduction of the MinFNR. MinFNR aims to minimize FNR and optimize computational resources, making it particularly suitable for high-risk applications. The proposed algorithm strategically combines data-level, algorithmic-level, and hybrid-level approaches. The ensemble algorithm is capable of improving the results of actual cases.

1.4. Organization

The rest of the paper is structured as follows. The next section describes the related work, exploring various approaches and algorithms to address imbalanced datasets. Section 3 provides background information and details the Set Covering Problem (SCP). In Section 4, the MinFNR ensemble algorithm is proposed, accompanied by a practical implementation. Results are discussed in Section 5, showcasing the algorithm's effectiveness across diverse datasets in terms of different approaches, performance metrics, and the MinFNR algorithm itself. Section 6 engages in a broader discussion on the implications of minimizing false negatives and the role of MinFNR in enhancing the credibility of machine learning applications in real-world scenarios. The final section offers concluding remarks.

2. Related work

This section addresses the algorithms designed to handle imbalanced datasets. An imbalanced dataset refers to a data collection in which the instances of one class are substantially fewer than those of another, resulting in a disproportionate representation of the various classes within the dataset. This kind of dataset is common in real-world activities, such as fraud detection, cancer diagnosis, spam detection in email, network intrusion detection, predictions of natural disasters, anomaly detection in manufacturing, insurance, pollution detection, remote sensing (land mine, underwater mine), and so many other cases.

In fraud detection, which typically involves a binary variable, the more significant proportion of transactions is expected to be legitimate, with a small fraction being fraudulent. This smaller subset of fraudulent transactions typically accounts for less than 5 % of the observations. When dealing with imbalanced datasets, a significant problem arises as the algorithm perceives the minority class, which contains vital data, as noise data. As a result, the classifier tends to neglect the minority class. There are several techniques in ML to deal with the misclassification of minority classes and false negatives. Haixiang et al. (2017), Zhu et al. (2018), and Gao et al. (2020) share a classification system that categorizes techniques into Data Level and Algorithmic Level. Additionally, Gao et al. propose a Hybrid Level combining other methods, as shown in Fig. 2.

The Data Level approach involves performing a pre-processing step to balance the dataset and reduce the negative impact on the minority class. Haixiang et al. (2017) describe data balancing techniques, including under-sampling and over-sampling, and both methods are applied combined and defined as the hybrid method. Haixiang et al. (2017) also include feature selection, which excludes features from the dataset.

Under-sampling deals with the random deletion of observations of the majority class, while over-sampling is about creating multiple copies of observations of the minority class. Both methods have disadvantages: under-sampling can discard instances of potentially valuable data, while over-sampling can increase the probability of overfitting.

A new technique to address the imbalanced dataset was published: SMOTE, Synthetic Minority Over-sampling Technique (Chawla et al., 2002). The SMOTE method creates synthetic examples of minority

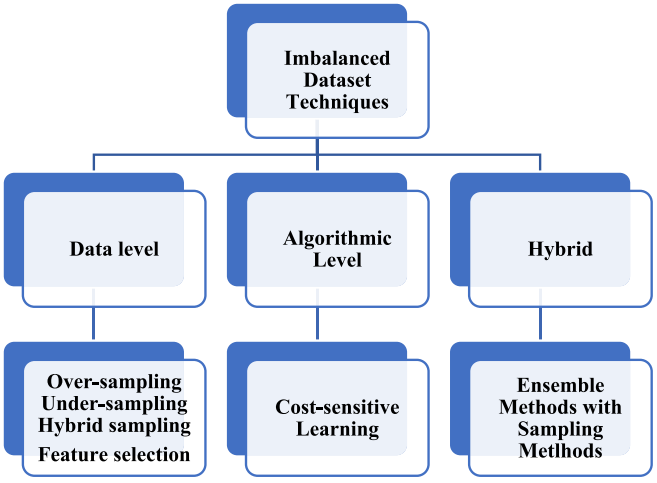


Fig. 2. Imbalanced dataset techniques.

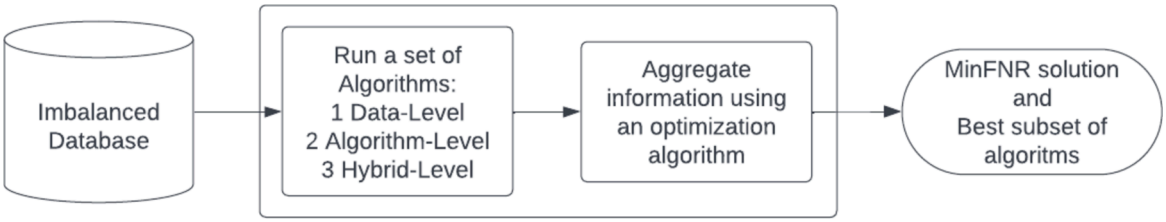


Fig. 1. The general schema of this work.

classes by incorporating information from minority and majority classes to generate a balanced dataset. The objective is to maintain the underlying pattern of the minority class in an adjusted training dataset to achieve balance and prevent algorithmic bias towards the majority class. This transformation creates a new dataset that preserves the pattern from the interest class and avoids the bias of the majority class.

Since the publication of the SMOTE technique 15 years ago, researchers have created over 85 SMOTE extensions, as (Fernández et al., 2018) reported. SMOTE SLS or Safe-Level-SMOTE (Bunkhumpornpat et al., 2009) is an extension of carefully selected samples of minority instances along the same line with different weight degrees and synthesized new minority class instances around the more significant safe level. Another SMOTE extension is the Density-Based Synthetic Minority Over-Sampling Technique, DBSMOTE (Bunkhumpornpat et al., 2012), which uses density-based clusters to over-sample an arbitrarily shaped cluster.

Although SMOTE is primarily an oversampling method, its Python and R implementations offer hyperparameters that adjust the levels of oversampling, undersampling, and the ratio of generating new samples that resemble those in the minority class, allowing customization to the unique attributes of the dataset. So, these implementations could be used as a hybrid method.

Another data-level approach is Random Over Sampling Examples (ROSE) (Lunardon et al., 2014). With a smoothed bootstrap approach, this technique forms artificially balanced samples that help estimate and evaluate a binary classifier's accuracy in a highly imbalanced dataset.

The Algorithmic Level offers a solution with an optimized or modified algorithm that considers specific characteristics of class imbalance scenarios without altering the dataset during the process. Cost-sensitive learning is a method that modifies the algorithm to consider the costs associated with a misclassifying of class. This method could contemplate the issues related to an imbalanced dataset: the low percentage of the minority class and the difference in relevance of false positives and false negatives in this context. Cost-sensitive learning considers the costs of prediction errors and seeks to reduce the cost with a cost matrix. Table 1 shows the ratio 1:500.

The value in the Cost Matrix represents the cost of misclassifying, and the higher cost is the false negative. The notation C(i,j) indicates the cost. The total cost is given by:

$$\text{Total Cost} = \text{Cost}(0,1) \times \text{FalseNegatives} + \text{Cost}(1,0) \times \text{FalsePositives}$$

For example, in a fraud context, the cost of an actual fraud committed without a fraud prediction is much higher than a predicted fraud that did not take place. This significant difference between costs highlights the importance of minimizing FN and eventually neglecting FP. The cost-sensitive algorithms are modified by adding a variable weight to reduce the FN. For Logistic Regression, the algorithm is modified to a weight model error by class weight to adjust coefficients by an optimization algorithm that minimizes negative log-likelihood (Brownlee, 2020).

In cost-sensitive processing, each class (or instance) receives an automatic classification cost, and the problem is to minimize the total cost of misclassification. For an imbalanced dataset, the weighting multiplies the cost function. In defining a higher weight for the minority class, logistic regression weighting can balance the effect of each class in the training process and reduce the issue of misclassification of the majority class. The cost-sensitive Decision Tree algorithm is a modified decision tree with a splitting criterion considering the misclassification costs. The algorithm splits by minimizing the weighted sum of misclassification costs for each class.

Table 1
Cost-sensitive confusion matrix.

	Actual Negative	Actual Positive
Predicted Negative	(True Negative, TN) Cost(0,0) = 0	(False Negative, FN) Cost(0,1) = 500
Predicted Positive	(False Positive, FP) Cost(1,0) = 1	(True Positive, TP) Cost(1,1) = 0

Hybrid Methods unify two or more techniques from the Data Level and Algorithmic Level to develop a cohesive strategy. These methods may involve an ensemble approach that integrates multiple prediction techniques from both levels, making predictions more robust and accurate than any individual model. Bagged decision trees with random under-sampling is a method that implements a modified bagged decision tree with random under-sampling of the majority class before fitting each decision tree. On the other hand, Random Forest with Bootstrap Class Weighting is a modified class distribution by weighting applied for each bootstrap sample to fit a decision tree.

In 2009, Xu-Ying Liu introduced Easy Ensemble (Liu et al., 2009), which combines bagging and boosting algorithms. This ensemble algorithm aims to reduce the issue of ignoring many majority-class samples, and it randomly selects samples from the majority class and all samples from the minority class. Then, a model could be fit by this dataset. The algorithm repeats this process multiple times and uses the average prediction of these repetitions as the final result.

3. Background information

This section provides background information on the field of Combinatorial Optimization. Talking about "easy" and "hard" problems is common in combinatorial optimization. The easy problems can be solved in polynomial time, also known as problems of class P. There is an efficient algorithm for solving easy problems. One classic example of an easy problem in combinatorial optimization is the Minimum Spanning Tree problem.

On the other hand, no efficient deterministic algorithm is known to solve hard problems in polynomial time. Therefore, they are classified in the NP class, i.e., nondeterministic polynomial time (Garey & Johnson, 1979). This research employs NP-hard problems, including the Set Covering Problem (SCP).

SCP is an NP-hard problem in which, given a collection of elements, the SCP aims to find the minimum number of sets that cover all of these elements (Wolsey, 2021).

The SCP arises in various contexts. For instance, the facility location problem involves choosing a few locations to open facilities that efficiently serve a given set of regions. Similarly, in network design, the SCP pertains to selecting the fewest network links required to guarantee the connectivity of all nodes.

Given a universe $U = \{A_1, A_2, \ldots, A_n\}$ with N elements and a collection of subsets of U, $S = \{S_1, \ldots, S_m\}$, the goal is to find a subset of S that covers all elements in U. Each subset Si contains a subset of the elements from the universe. For example, $S1 = \{A_1, A_2, A_3\}$, $S_2 = \{A_3, A_4\}$, $S_3 = \{A_2, A_5\}$. The goal of the SCP is to find a minimum-size subset cover, which is a subset of S, named in this context as C, such that the union of the selected subsets in C covers the entire universe U. A binary decision variable x(j) is introduced for each subset Si in S. If $x(j) = 1$, it means that S_i is selected for the cover; if $x(j) = 0$, it is not selected. The objective function is to minimize the total number of selected subsets, which can be expressed as follows.

$$min \sum_{j=1}^{m} x_j$$

Constraints: The constraints ensure that the selected subsets cover the entire universe U; for each element A_i in the universe, there must be at least one subset containing A_i, expressed as follows. For each A_i in U: $\sum_{j=1}^{n} x_j S_j \geq 1$, where S_j contains A_i. The summation is taken over all subsets S_i that contain the element A_i. The constraint ensures that at least one subset covers each element in the universe.

A specific approach was proposed (Johnson, 1974) and (Lovász, 1975) and evolved in (Chvatal, 1979) into a Greedy Heuristic for the Set

Covering Problem. Another efficient algorithm is Branch-and-Bound, an optimization method that breaks problems into smaller sub-problems, using bounding to eliminate sub-problems that cannot contain the optimal solution. In this algorithm, branches are developed as long as their cost is promising, and it explores all possible solutions to find the lowest total cost.

In this work, a Branch-and-Bound implementation in Python uses auxiliary functions like bypassing the branch for infeasible solutions and the next vertex to create new subsets. This algorithm consistently delivers the optimal solution by thoroughly evaluating all potential solutions and selecting the one with the lowest cost, offering a logical proposition.

4. Ensemble algorithm proposal

This topic presents an ensemble algorithm classified as a hybrid-level approach. The hybrid approach involves training the training dataset using a combination of algorithms of the Data Level, Algorithm Level, and Hybrid approach previously defined. After fitting the model to the training dataset, the study evaluates its performance on the test dataset to find the performance measures Area Under the ROC Curve (ROC-AUC) and FNR.

4.1. The relevance of FNR

AUC and FNR are two relevant metrics used to assess the performance of a model in binary classification tasks with imbalanced datasets. In (Kou et al., 2004), the authors relate metrics for fraud detection techniques.

A confusion matrix is used to assess the performance of a machine learning algorithm by presenting the predicted values in comparison with the actual values in a test dataset. The predicted class labels and the actual class labels define four cells in the confusion matrix: True Positive (TP) is the number of correctly predicted positive cases; False Positive (FP) is the number of incorrectly predicted positive cases; True Negative (TN) is the number of correctly predicted negative cases, and False Negative (FN) is the number of incorrectly predicted negative cases. The confusion matrix makes it possible to show the model's performance by delivering the correct or incorrect predictions and which classes are being confused with each other.

The ROC curve (receiver operating characteristic curve) represents the performance of a binary classifier system that differentiates between two classes, typically labeled as positive and negative. The ROC curve is the result of plotting the True positive rate (TPR) and false positive rate (FPR). The AUC can express the model's performance of a binary classifier system (Mandrekar, 2010) and indicates that an AUC value of 0.5 signifies no discrimination. In contrast, 0.7 to 0.8 is considered acceptable, 0.8–0.9 is considered excellent, and a value exceeding 0.9 is considered exceptional.

FNR is the proportion of instances of positive class incorrectly predicted as negative, i.e., the number of false-negatives divided by the sum of True-Positives (TP) and False-Negatives (FN), and given by FNR = FN/ (FN + TP). FNR is a significant metric in evaluating the performance of a binary classification model, especially when the cost of a false negative is high.

False Negatives and False Positives have different importance in a fraud dataset. False Positive represents someone predicted as a fraud who has committed no crime. In False Negative, the error is worse; the algorithm's prediction indicates that the individual is not fraudulent when he committed fraud in real life.

These different errors have different relevance, and this is an issue that ML must address. In imbalanced scenarios, the cost of a false negative may be higher than a false positive. Therefore, minimizing the FNR is essential to ensure the model makes accurate predictions for the interest class, and the cost of FPR can be neglected.

FNR is closely related to the Recall, also known as Sensitivity or True Positive Rate (TPR), TPR = TP/ (FN + TP). The FNR is the complement of Recall, meaning that these indicators are inversely related. On the other hand, precision is another measure defined as the ratio of true positives to the sum of true positives and false positives, calculated as Precision = TP / (FP + TP). Precision is relevant when the cost of false positives is high, meaning that minimizing false positives is more important than minimizing false negatives.

This study addresses highly and extremely imbalanced datasets, where the relevant class represents less than 5 % and less than 1 % of the data, respectively. In these contexts, false negatives are considered more critical than false positives.

4.2. The ensemble algorithm MinFNR

As already stated, the primary goal of this study is to decrease the occurrence of False Negatives. The proposed ensemble algorithm is based on a set of classification algorithms A={A_1, A_2, …, A_m}. Each algorithm generates a unique solution with specific False Negatives and a distinct FNR. The computational results that make the correct predictions are aggregated into groups of true positives TP= {TP_1, TP_2, …, TP_n}. The information is gathered in a matrix M= {(x, y): x ∈ TP, y ∈ A}, with dimension n × m, defined by:

$$M_{x,y} = \begin{cases} 1 \text{ if } y \in A \text{ predicts } x \in TP \\ 0 \text{ otherwise} \end{cases}$$

The generation of the matrix M is obtained by running each classification algorithm of set A using the standard training set and a test set of dataset D. First, each algorithm of set A runs the training set of dataset D to train the models. Next, each algorithm of set A runs the test set to evaluate its performance. The correct predictions made by each algorithm are aggregated into matrix M, with each row representing a true positive TP and the corresponding algorithm set A that made the correct prediction.

The proposed algorithm that searches the minimum FNR is coined MinFNR. The proposed algorithm, MinFNR, benefits from the solutions of the different algorithms. The ensemble algorithm combines the correct predictions of different algorithms to reduce the FNR.

The algorithm MinFNR takes as input a dataset D set of algorithms Ai and aims to find two goals. The algorithms' outputs are the best subset of algorithms from set A and the value of the minimum FNR (false negative rate). The algorithm can be scratched as follows:

Algorithm MinFNR

Input: Dataset D, Set of algorithms A;
Output: the best subset of A, the value of the minimum FNR;
1. Given dataset D, generate a two-way table T
1.1 Run for each y ∈ A with a training set of dataset D;
1.2 Run for each y ∈ A with the test set of dataset D;
1.3 Aggregate the correct predictions into matrix M;
2. Find the best subset of A;
3. Calculate the value of the minimum FNR.

In order to clarify the algorithm, a running example is presented. Firstly, it is necessary to analyze the predictions of various algorithms for identifying fraud instances (i.e., class = 1). Once the models for each algorithm were trained using the training data, they were utilized to generate predictions.

Table 2 shows an example of the correct predictions of the algorithms {A_1, A_2, A_3, A_4}. The computational results that make the correct predictions were aggregated into groups of true positives {TP_1, TP_2, TP_3, TP_4, TP_5}. In addition to matrix M, the Counter indicates the frequency of the correct predictions in the training dataset. TP_1 means that all algorithms incorrectly predicted four occurrences. TP_2 means that the algorithms A_1 and A_2 correctly predicted one fraudulent element. TP_4 means the algorithm A_4 found two frauds. Finally, TP_5 shows that all

Table 2
Correct Prediction Example.

True Positive	Actual Value	Algorithm predictions for actual value = 1				Counter
		A_1	A_2	A_3	A_4	
TP_1	1	0	0	0	0	4
TP_2	1	1	1	0	0	1
TP_3	1	1	0	1	0	1
TP_4	1	0	0	0	1	2
TP_5	1	1	1	1	1	54

algorithms found 54 instances of fraud in the dataset, i.e., all algorithms accurately predicted the occurrence of fraud.

5. The best subset of algorithms

In order to find the best subset of A, a new reduced matrix B is extracted from matrix M by removing line TP_1 since no algorithm covers TP_1; in this running example, matrix B is as follows:

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

This study uses the matrix $[B_{i,j}]$ with i = 1, …, n lines or aggregated correct predictions, and j = 1, …, m columns or algorithms. The matrix $[B_{i,j}]$ is input in the Minimum Set Covering Problem, which aims to find the minimum number of columns that cover all lines. All constraints (or lines) must be covered at least once. The Integer Linear Programming formulation of the Minimum Set Covering problem is stated as follows:

$$\text{Minimize} f = \sum_{j=1}^{m} x_j$$

$$subject\ to \sum_{j=1}^{m} B_{i,j}.x_j \geq 1, \quad \forall i$$

and $x_j \in \{0,1\}, \quad with\ j = 1,\ldots,m$

In the example, running only the algorithms A_1 and A_4, all the fraudulent elements can be found. In other words, the best subset of the algorithms is {A_1, A_4}.

6. Minimum FNR value

Given matrix M, the FNR of an algorithm k can be calculated based on the correct predictions of the algorithm k ∈ A and vector Counter. The FNR^k is given by:

$$FNR^k = \frac{FN^k}{FN^k + TP^k} = \frac{\sum Counter_i - \sum A_i^k.Counter_i}{\sum Counter_i}$$

Given the aggregated table T and the best subset, the minimum FNR is calculated. The minimum FNR is obtained by the best algorithm(s) with the lowest rate of false negatives.

Table 3
Computing MinFNR.

True Positive	Actual Value	A_1	A_4	{A_1, A_4}	Counter
TP_1	1	0	0	0	4
TP_2	1	1	0	1	1
TP_3	1	1	0	1	1
TP_4	1	0	1	1	2
TP_5	1	1	1	1	54
TP=	62	56	56	58	
FN=	0	6	6	4	
FNR=	0.00 %	9.68 %	9.68 %	6.45 %	

Table 3 shows the reduced table T for the subset of the algorithms {A_1, A_4}. The FNR for A_1 and A_4 is 9.68 %. The combined solutions decrease the FNR to 6.45 %, corresponding to the MinFNR.

The algorithm MinFNR achieves two aligned goals: the minimum subset of algorithms and the minimum FNR by combining solutions. The best subset of algorithms can be found using a heuristic or an exact Set Covering algorithm. Knowing the best subset, we can dispense running all the algorithms in the future.

6.1. Extended formulation

We can measure the computational running times since running the m algorithms is necessary. Computational times are considered a cost c ($_j$) that is intended to be minimized. Considering computational times, the extended formulation is as follows:

$$\text{Minimize} f = \sum_{j=1}^{m} c_j.x_j$$

$$subject\ to \sum_{j=1}^{m} B_{i,j}.x_j \geq 1, \quad \forall i$$

and $x_j \in \{0,1\}, \quad with\ j = 1,\ldots,m$

This section provides a detailed proposal and description of an ensemble algorithm called MinFNR for binary classification tasks with imbalanced datasets, specifically focused on minimizing FNR for fraud detection. This algorithm combines the strengths of multiple classification algorithms to reduce the FNR and improve the model's performance. This comprehensive proposal aims to improve the performance of fraud detection models by leveraging ensemble techniques to reduce false negatives, which are costly in this context.

7. Computational results

Decisions regarding the computational environment, datasets, algorithms, and performance metrics are essential for accurate computational results.

Regarding the computational environment, the experiments are conducted on an Intel Core i7 11700K 4.9 GHz processor and 64 GB of RAM, using a Windows 11 Operating System. For the programming software, two languages are used: Python with packages Pandas 1.4.4, NumPy 1.21.5, Scikit-Learn 1.1.3, and Imbalanced-learn 0.9.1; and R version 4.2.2, RStudio version 2023.09.0 + 463, with packages SmoteFamily 1.3.1 and ROSE 0.0-4.

The following subsections describe the datasets, the algorithms, and the performance measures based on the FNR.

7.1. Datasets description

In this study, five distinct datasets are included, each described below. The selection of these datasets was driven by their highly imbalanced nature, which is the primary focus of this research. A detailed overview of the datasets, including the number of attributes, lines, and the imbalance percentage, is shown in Table 4.

The first dataset, "Air Pollution Norwegian" (Air Pollution Norwegian, 2004), is accessible at https://lib.stat.cmu.edu/datasets/. It

Table 4
Statistical information about the datasets.

Name	# Attributes	# Lines	Imbalanced (%)
Air Pollution Norwegian	7	500	2.20 %
Bioassay Burnham	154	9,955	0.27 %
Corruption	27	303,036	0.14 %
Credit Card Fraud Detection	29	284,807	0.17 %
SID-200 K-50	50	200,000	0.10 %

originated from a study investigating the relationship between road air pollution, traffic volume, and meteorological variables in Norway. The Norwegian Public Roads Administration collected the data. The response variable comprises hourly values of the NO2 (particle) concentration logarithm, measured at Alnabru in Oslo, Norway, from October 2001 to August 2003. The predictor variables include the number of cars per hour, temperature, wind speed, wind direction, the hour of the day, and day number.

The second dataset, "Bioassay Burnham AID456″ González-Fabra et al. (2017), is a primary screen assay from the Burnham Center for Chemical Genomics. It assesses the inhibition of Tumor Necrosis Factor-alpha (TNFa-induced) Vascular Cell Adhesion Molecule-1 (VCAM-1) cell surface expression.

The third dataset, referred to as "Corruption risk assessment in a public administration (Vasconcelos & Cavique 2022), is mentioned in this paper as Corruption and is available at https://data.mendeley.com/datasets/crpdknzswh/2. This dataset is extensively detailed in the Data in Brief Journal. It was created by integrating data from eight different systems of the Brazilian federal government and the Federal District, focusing on actual data from civil servants and military personnel.

The fourth dataset, "Credit Card Fraud Detection," is sourced from Kaggle and can be found at https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud Andrea (2017). It features transactions over two days, with all numerical variables resulting from PCA transformation, except for 'Time' and 'Amount.'.

The final dataset, 'SID-200 K-50′, is a synthetic imbalanced dataset generated using specialized code designed to create datasets with highly skewed class distributions. A comprehensive explanation of this process can be found in the section titled "Choose Better Metrics, Balance Skewed Classes, and Apply Cost-Sensitive Learning" (Brownlee, 2020).

7.2. Algorithms description

In this study, 15 algorithms are selected to deal with imbalanced datasets, as shown in Table 5. The algorithms are divided into three groups: data level, algorithmic level, and hybrid level. As a reflection of best practices in machine learning when dealing with imbalanced datasets, all algorithms underwent a ten-fold cross-validation repeated three times, guaranteeing consistent class proportions across all folds.

In the Data Level approach (A_1–A_7), SMOTE, SMOTE extensions, and ROSE are used to create a balanced dataset for model training and fitting, thereby eliminating bias from the majority class of the original dataset. The outcomes are obtained by applying these techniques for Logistic Regression.

In the Algorithmic Level approach (A_8–A_{12}), an advanced algorithm is utilized, featuring a range of tailored parameters designed for effective cost-sensitive learning without data sampling, except A_{11} and A_{12}, with no skills available for a benchmark. The dataset is evaluated using Logistic Regression, Artificial Neural Network, Random Forest, Decision Tree, and Support Vector Machine (SVM). However, only the top-performing models are showcased.

Several ensemble algorithms were tested regarding the Hybrid Level, including Easy Ensemble, Bagged Decision Trees, Random Forest with Bootstrap Class Weighting, and various modified versions. Only the top-performing models are showcased (A_{13}–A_{15}). Finally, algorithm A_{16} corresponds to the proposed ensemble approach.

Table 5
Data Level, Algorithmic Level, Hybrid, and Proposal.

ALGORITHMS	
Data Level	A_1 SMOTE ADASYN + RegLog
	A_2 SVMSMOTE + RegLog
	A_3 BorderlineSMOTE + RegLog
	A_4 SMOTE Random Under-sampling (Over-sampling 0.7, Under-sampling 1)
	A_5 ROSE + RegLog Under-sampling / Over-sampling (link Cauchit)
	A_6 ROSE + RegLog Under-sampling / Over-sampling (link Logit)
	A_7 ROSE + RegLog Under-sampling
Algorithmic Level	A_8 Weighted Logistic Regression {0:1, 1:578}
	A_9 Weighted Logistic Regression {0:1, 1:1000}
	A_{10} Weighted Logistic Regression Heuristic
	A_{11} Logistic Regression (no skills)
	A_{12} Decision Tree (no skills)
Hybrid Level	A_{13} Ensemble Algorithm − Easy Ensemble
	A_{14} Bagged Decision Trees with Random Under
	A_{15} Random Forest with Bootstrap Weighting
	A_{16} Our Ensemble Algorithm (MinFNR)

7.3. Combining the correct prediction

Each of the 15 algorithms is executed on every dataset. The goal is to generate a matrix that records the number of accurate predictions made by each algorithm. This matrix will then be used to run the proposed algorithm. The outcome of the algorithm MinFNR is a subset of A, and the new algorithm A_{16} is the union of the given subset. Combining these algorithms, we develop a more compelling hybrid algorithm specifically designed for handling imbalanced data.

For example, Table 6 displays the correct predictions of the Corruption Dataset algorithms for True Positive. The most suitable subset of algorithms to minimize FNR can be found by running multiple algorithms simultaneously. The best algorithm subset is selected through a set covering problem formulation. In this study, the best subset of A equals {A_{12}, A_{14}, A_{15}}. The algorithm A_{16} is the union of A_{12}, A_{14} and A_{15}. In future research with similar datasets, only three algorithms should be run instead of 15.

All algorithms find 195 fraudulent elements (TP_8), although the combination of the algorithms can find 128 more frauds (TP_2-TP_{15}). Our ensemble algorithm MinFNR (A_{16}) takes advantage of the previous ones and retrieves the excellent FNR of 1.89 %. It is interesting to note that A_{14} with a poor FNR contributes to the best solution.

7.4. Performance measures

This section delves into the outcomes of both the employed algorithms (A_1-A_{15}) and the proposal algorithm (A_{16a} − A_{16b}), presenting performance metrics such as AUC, FNR (%), and the time of the training in seconds for the most effective algorithms, are presented in Table 7. These results, categorized into three levels across five datasets, offer insights into the algorithms' performance. The assessment of each algorithm's outcomes involves diverse approaches and critical indicators, with the subsequent graphic illustrating the average values per level for each indicator.

Despite a wide range of FNR values (0 to 100 %), consistent values can be obtained for developing an efficient proposal algorithm (A_{16a} − A_{16b}). The proposal ensemble algorithm MinFNR offers two distinct

Table 6
Correct Predictions of the Corruption Dataset.

True Positive	A_{12}	A_{14}	A_{15}	$A_{16} = A_{12} \cup A_{14} \cup A_{15}$	Counter
TP_1	0	0	0		2
TP_2	0	0	1	1	4
TP_3	0	1	0	1	8
TP_4	0	1	1	1	25
TP_5	1	0	0	1	2
TP_6	1	0	1	1	11
TP_7	1	1	0	1	76
TP_8	1	1	1	1	195
TP=	39	19	88	321	323
FN=	284	304	235	2	
FNR=	12.07 %	58.88 %	27.24 %	1.89 %	

Table 7
Computational results.

Datasets		Air Pollution Norwegian			Bioassay − Burnham			Corruption			Credit Card Fraud Detection			SID-200 K-50		
Algorithms		AUC	FNR (%)	time (sec)	AUC	FNR (%)	time (sec)	AUC	FNR (%)	time (sec.)	AUC	FNR (%)	time (sec)	AUC	FNR (%)	time (sec)
Data Level	A_1 SMOTE ADASYN + RegLog	0.956	37.50	0.0	0.993	77.78	5.3	0.702	66.56	9.9	0.980	27.59	21.2	0.966	32.67	0.3
	A_2 SVMSMOTE + RegLog	0.954	87.50	0.0	0.993	77.78	6.2	0.702	87.93	8.9	0.979	87.53	16.9	0.966	88.00	0.3
	A_3 BorderlineSMOTE + RegLog	0.904	100.00	0.0	0.66	77.78	0.8	0.507	100.00	1.2	0.958	99.73	14.6	0.892	100.00	0.2
	A_4 SMOTE Random Under (Over 0.7 Under 1)	0.918	75.00	0.0	0.560	100.00	0.8	0.546	100.00	1.1	0.977	40.85	14.0	0.943	92.67	0.2
Algorithmic Level	A_8 Weighted Logistic Regression {0:1, 1:578}	0.915	0.00	0.0	0.659	25.93	2.8	0.667	57.89	4.8	0.980	8.22	9.3	0.944	12.00	0.2
	A_9 Weighted Logistic Regression {0:1, 1:1000}	0.924	0.00	0.0	0.656	25.93	3.0	0.667	48.61	5.6	0.979	8.22	9.8	0.944	2.67	0.2
	A_{10} Weighted Logistic Regression Heuristic	0.914	0.00	0.0	0.674	25.93	2.5	0.667	33.13	4.8	0.980	6.90	8.6	0.944	2.67	0.2
	A_{11} Logistic Regression (no skills)	0.932	75.00	0.0	0.765	100.00	0.9	0.688	100.00	1.3	0.975	40.85	14.1	0.919	92.67	0.2
	A_{12} Decision Tree (no skills)	0.546	0.00	0.0	0.521	25.93	0.2	0.563	12.07	2.2	0.876	0.00	10.6	0.59	0.00	7.3
Hybrid Level	A_{13} Ensemble Algorithm − Easy Ensemble	0.913	0.00	0.3	0.631	25.93	0.4	0.718	28.48	1.9	0.978	0.27	1.7	0.949	0.00	1.5
	A_{14} Bagged Decision Trees with Random Under	0.900	0.00	0.0	0.632	25.93	0.1	0.729	5.88	2.5	0.972	3.18	1.1	0.955	2.67	1.2
	A_{15} Random Forest with Bootstrap Weighting	0.790	0.00	0.0	0.519	44.44	0.1	0.588	27.24	1.7	0.928	6.10	5.9	0.606	20.67	4.4
	A_{16a} *Our Ensemble Algorithm (MinFNR) without cost*	0.956	0.00	0.0	0.993	22.22	1.0	0.729	1.89	6.4	0.980	0.00	2.8	0.966	0.00	8.7
	A_{16b} *Our Ensemble Algorithm (MinFNR) with cost (time)*	0.956	0.00	0.0	0.993	22.22	1.0	0.729	1.89	6.4	0.980	0.00	2.8	0.966	0.00	7.5

approaches: A_{16a} disregards cost, while A_{16b} considers costs, where the costs are defined as the execution times of algorithms (A_1-A_{15}).

Our MinFNR (A_{16a} − A_{16b}), with high AUC, is highly consistent in its ability to discriminate between positive and negative classes. The shallow FNR values across datasets indicate their ability to identify positive instances correctly. These algorithms are practical in scenarios where avoiding false negatives is crucial.

The ensemble algorithm MinFNR shows high performance in AUC and FNR. The AUC is maximized, and the FNR is minimized, outperforming all the other algorithms with a reduced computational time.

7.4.1. Performance measures by levels

The charts in Fig. 3 illustrate the behavior of crucial indicators across different levels. The average value was obtained from algorithms at each level (data, algorithmic, hybrid).

A high AUC value expresses that the model can discriminate between the positive and negative classes. At the Data level, the AUC presents a better average than at the Algorithmic Level, and it decreases at the Hybrid Level.

The FNR graphs exhibit comparable patterns to the AUC graphs at various levels, as indicated by a Pearson correlation coefficient of 0.99, yet convey different meanings. A decrease in AUC signifies a decline in performance, whereas a decrease in FNR indicates a performance improvement. A low rate of false negatives reveals the optimal outcome.

The graphic of execution time average along the levels shows different patterns according to the datasets. A relevant variable not shown in this graphic that has an influence is the number of lines of the dataset; this attribute presents a moderate correlation regarding the execution time (0.68). The smallest dataset (Air Pollution Norwegian, 500 lines) showed almost no variation, but the most significant (Corruption − 303,036 lines, and Credit Card Fraud Detection − 284,807) presents the most extensive execution time.

In the context of the specified approaches, concerning the Data Level approach, the performance of all outcomes generated with the SMOTE Technique (A_1-A_4) surpassed that of the results obtained using the ROSE Technique (A_5-A_7). As a result, the findings derived from the ROSE Technique were excluded.

7.4.2. FN rates per dataset

Fig. 4 depicts the FNR performance for individual algorithms (A_1-A_{15}) and our ensemble algorithm (A_{16a} and A_{16b}) for each dataset by the distance measured by radius from the zero pointer.

In particular visual representations, higher values observed in A_1, A_2, A_3, A_4, and A_{11} indicate a less favorable performance regarding the FNR. Conversely, other algorithms exhibit values close to zero or equal to zero, reflecting superior performance.

In the spectrum of approaches, it is evident that the data-level algorithms (A_1-A_4) consistently deliver the least favorable FNR performance across all datasets. In contrast, the hybrid-level approach (A_{13}-A_{15}) algorithms consistently achieve the most favorable FNR values, signifying superior performance for the selected datasets than in individual applications.

Finally, our ensemble algorithm consistently outperforms the individual algorithms, demonstrating a superior FNR performance and attaining values lower than those of the individual algorithms.

7.5. Algorithms subset

The preceding subsections discuss the performance measures of the MinFNR algorithm. This subsection details the minimum subset of algorithms provided by MinFNR. The SCP algorithm was applied to identify each dataset's MinFNR results (A_{16a} and A_{16b}). On the other hand, the columns (or algorithms) selected from the SCP matrix constitute the subset of chosen algorithms. The selected algorithms are shown in Table 8 using boolean expressions.

For instance, seven algorithms cover all the true positives (TP) in the

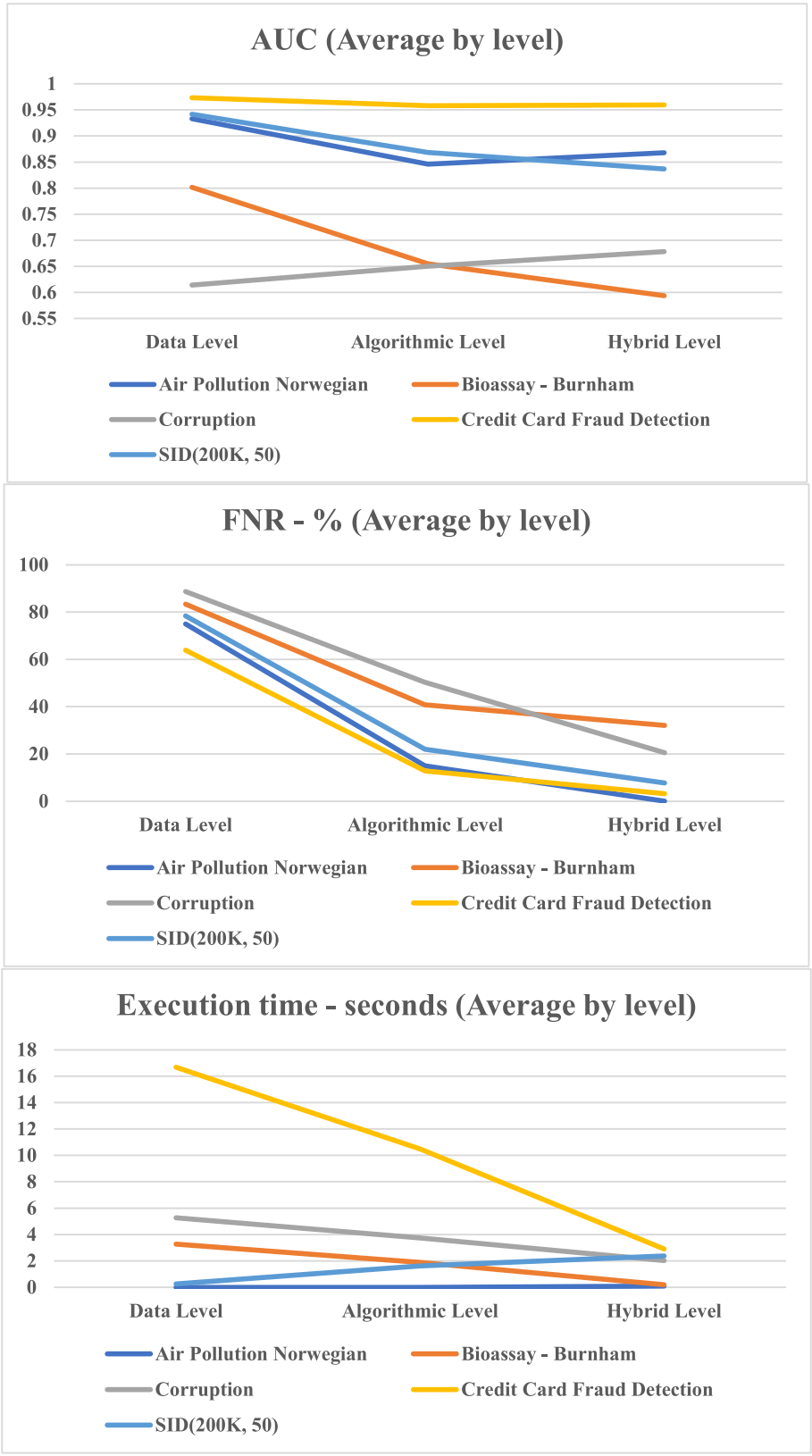


Fig. 3. Performance measures by levels.

Air Pollution Norwegian dataset. Meanwhile, dataset corruption needs three algorithms to cover all true positives.

The next step was to obtain the reduced proposition by combining the Boolean expressions provided by SCP of each dataset using the AND operator and subsequently simplifying this combined expression using available methods, such as Boolean Algebra, De Morgan's Theorems, or Karnaugh Maps.

The reduced proposition of the boolean expressions corresponds to

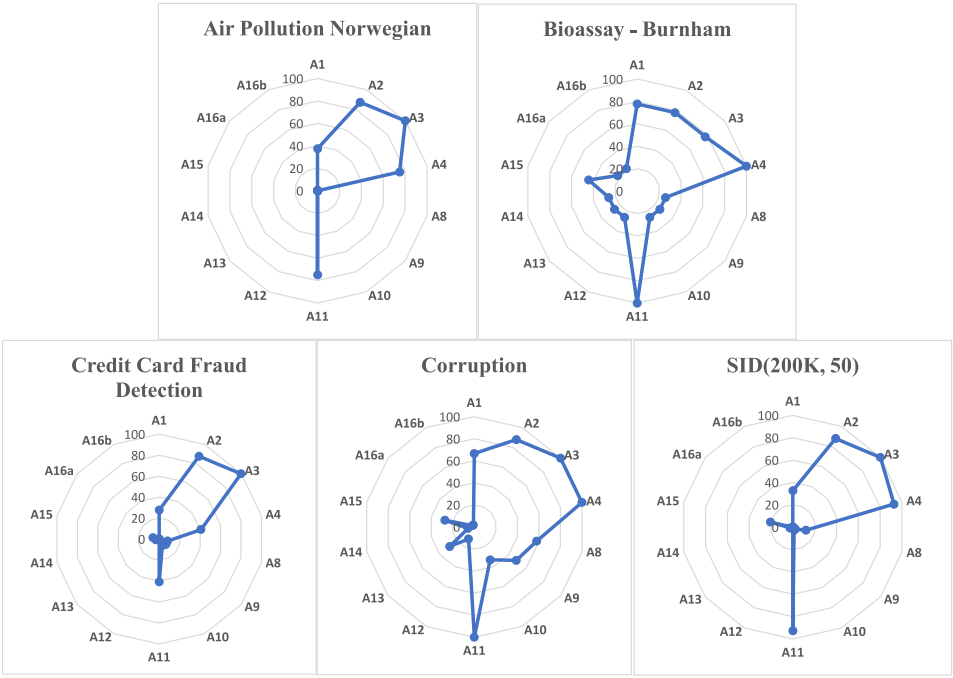


Fig. 4. The false negative rate is applied to all datasets and all algorithms.

Table 8
Selected algorithms provided by SCP.

Dataset	Unweighted (cost = 1) A_{16b} A_{16a}	Weighted (costs ≠ 1) A_{16b}
Air Pollution Norwegian	$(A_8 \vee A_9 \vee A_{10} \vee A_{12} \vee A_{13} \vee A_{14} \vee A_{15})$	$(A_8 \vee A_9 \vee A_{10} \vee A_{12} \vee A_{13} \vee A_{14} \vee A_{15})$
Bioassay Burnham	$(A_1 \vee A_3) \wedge (A_8 \vee A_9 \vee A_{10} \vee A_{12} \vee A_{13})$	$(A_1 \vee A_3) \wedge (A_8 \vee A_9 \vee A_{10} \vee A_{12} \vee A_{13})$
Corruption	$A_{12} \wedge A_{14} \wedge A_{15}$	$A_{12} \wedge A_{14} \wedge A_{15}$
Credit Card Fraud Detection	A_{12}	$A_{13} \wedge A_{14}$
SID(200 K, 50)	$(A_{12} \vee A_{13})$	$(A_{12} \vee A_{13})$
Reduced proposition	$\mathbf{A_{12} \wedge A_{14} \wedge A_{15} \wedge (A_1 \vee A_3)}$	$\mathbf{A_{12} \wedge A_{13} \wedge A_{14} \wedge A_{15} \wedge (A_1 \vee A_3)}$

the subset of algorithms needed for all datasets. The reduced set includes four algorithms for unweighted solutions or five for weighted solutions instead of the original set of fifteen. An additional algorithm, A_{13}, is needed for the weighted formulation.

To build the optimal ensemble algorithm, the proposed MinFNR (A_{16a} and A_{16b}) incorporates algorithms from all levels: Data level (A_1 or A_3), Algorithmic Level (A_{12}), and Hybrid Level (A_{13}, A_{14}, A_{15}). While the data-level approach yields a low FNR value, it is crucial to consider its contribution to achieving the best overall result.

8. Discussion

Regarding the independence of the test subset, which ensures that it contains no data previously seen during training that allows for an unbiased evaluation of the models, we must clarify the differences between model selection and model combination.

In model selection, when comparing multiple models or variations of the same model, an intermediary subset, the validation subset, helps to select the best-performing model. Peeperkorn et al. (2024) identify the relevance and importance of partitioning the dataset in training, validation, and testing subsets. On the other hand, the model combination used in our approach involves aggregating the FNR predictions from multiple models to enhance the overall predictive accuracy beyond what any single model could achieve. The independence of the test subset is guaranteed for each algorithm. Our ensemble approach, where the SCP finds an ensemble solution that minimizes the FNR, merges the FNR solutions without changing the independence of the test subsets.

Area Under the Curve (AUC) is intensely used when evaluating the performance of binary classification models. However, False Negative Rate, FNR, becomes especially important when not correctly identifying positive cases (false negatives) could have serious consequences, such as fraud detection, medical diagnosis, security checks, and other fields.

In these fields, a false negative can have various significant repercussions. For instance, fraud detection can result in financial losses due to undetected fraudulent activities. In medical diagnostics, missing a diagnosis can delay treatments and worsen patient outcomes by not identifying severe health conditions. In airport security, overlooking a prohibited item or threat can pose a significant risk to public safety. In cybersecurity, false negatives allow attackers to breach systems undetected. In industries like automotive or aerospace manufacturing, it can lead to the distribution of faulty products, potentially causing serious harm. In the context of quality control, the risk of selling defective products increases. Not recognizing actual fires or smoke leads to slower responses and more extensive damage to fire detection systems. Finally, failing to locate a person in need can result in life-threatening situations in search and rescue operations.

Highlight a tendency in the artificial intelligence (AI) field to underestimate the consequences of false negatives. Although the repercussions of false positives are widely recognized due to the adverse outcomes they cause, the costs and impacts of false negatives often receive less attention. This oversight can lead to individual consequences and broader issues regarding public trust in AI systems.

In fraud detection, a False Positive is when a valid transaction is incorrectly flagged as fraud, leading to inconveniences like a temporarily suspended card. On the other hand, a False Negative is when the system misses a fraudulent activity, resulting in unauthorized account access or overlooked fraudulent transactions. For AI operators, it is easier to measure and address losses from false positives, but false negatives present distinct challenges.

In this study, the AUC is not the primary performance metric for addressing fraud detection with imbalanced datasets. While Knowles et al. (2023) highlight crucial concerns associated with the FNR, AUC

proves inadequate in resolving these issues. AUC is specifically designed to evaluate the overall discriminatory capacity of a classification model.

The goal of the MinFNR ensemble algorithm is to minimize the FNR. It uses classification algorithms to reduce false negatives while improving overall performance. The algorithm aggregates correct predictions from each algorithm, aiming to discover the minimum FNR. This approach minimizes computational time and resources by identifying the best-performing subset of algorithms. We consider the MinFNR algorithm as a foundational tool that aligns with Knowles' requirements (Knowles et al., 2023) and aims to enhance the credibility of AI.

9. Conclusions

This work explores the significant challenges arising from high or highly imbalanced datasets in the context of binary classification. Imbalanced classification refers to a challenge in machine learning where the target attribute in the training data is not represented equally. This imbalance can significantly skew the performance of classification models, leading to a situation where they perform well on the majority class but poorly on the minority classes. Imbalanced datasets are usual in many real-world applications, such as fraud detection, medical diagnosis, and spam detection, where one class might be substantially more frequent.

Several methods in machine learning are employed to address the issue of minority class misclassification and false negatives in imbalanced datasets. These techniques can be broadly categorized into three types: Data Level, Algorithmic Level, and Hybrid Level approaches. The proposed ensemble algorithm is classified as a hybrid approach.

Most algorithms overvalue accuracy and neglect false negatives (FN). False negatives are particularly costly in real-world applications like fraud detection, cancer diagnosis, network intrusion detection, forecasting natural disasters, identifying anomalies in manufacturing, insurance assessments, and numerous other domains. In intrusion detection, the difference between a false negative and a false positive is equivalent to a perfect crime and a false alarm.

The false negative rate (FNR) is a critical metric in assessing the effectiveness of binary classification models in this context. The distinction between false positives and false negatives is emphasized, underscoring that these errors carry different significance levels. In scenarios where the cost of false negatives outweighs that of false positives, minimizing FNR becomes paramount. This concept underscores the fundamental premise of this work.

The ensemble algorithm MinFNR is introduced to minimize the FNR in imbalanced datasets in the first place. A second goal is to identify the optimal subset of these algorithms that collectively minimize FNR. The MinFNR algorithm aligns with the primary focus of this paper by reducing false negatives while conserving computational resources.

MinFNR employs the optimization formulation of the Set Covering Problem (SCP) to find an ensemble solution that minimizes the FNR. Boolean expression simplifications are used to find a reduced subset of algorithms.

A comprehensive evaluation of the MinFNR algorithm is conducted within a real-world computational environment. Five datasets, ranging from environmental data to fraud detection scenarios, are used to assess the algorithm's performance. Fifteen algorithms across different levels are analyzed, and MinFNR consistently outperforms individual algorithms, confirming its potential.

The MinFNR ensemble algorithm emerges as a promising solution for mitigating the challenges posed by imbalanced datasets, particularly in industries where the cost of false negatives, such as fraud detection, is substantial. By leveraging a combination of algorithms and considering computational time costs, MinFNR improves predictive qualities and optimizes resource utilization, making it a compelling choice for high-stakes applications. MinFNR, as a foundational tool, addresses the critical need to minimize false negatives in scenarios where overlooking positive instances carries significant consequences, contributing to ongoing efforts to enhance the credibility and effectiveness of machine learning algorithms in real-world applications.

This study has made significant progress in tackling the challenges of FNR in imbalanced datasets in binary classification. However, there are still areas for future research and improvement, such as in scenarios where a high frequency of false positives can make the predictive application impractical. While the MinFNR algorithm is designed to minimize the False Negative Rate (FNR), future research could verify the frequency of false positives. Balancing these metrics would make the predictive application more practical and effective.

Declaration of Generative AI and AI-assisted technologies in the writing process

While preparing this work, the authors used ChatGPT and Grammarly to improve the grammatical quality of the previously written and submitted text. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

CRediT authorship contribution statement

Marcelo Vasconcelos: Conceptualization, Methodology, Investigation, Data curation, Visualization, Writing – original draft, Writing – review & editing, Software. **Luís Cavique:** Conceptualization, Supervision, Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to the data on the manuscript.

References

[dataset] Air Pollution Norwegian. Magne, Aldrin (2004, July 28). https://lib.stat.cmu.edu/datasets/NO2.dat. *Accessed March 27, 2024.*

Brownlee, J. (2020). Imbalanced classification with python - choose better metrics, balance skewed classes, and apply cost-sensitive learning. *Machine Learning Mastery, 463*.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-01307-2_43

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: Density-based synthetic minority over-sampling technique. *Applied Intelligence, 36*(3), 664–684. https://doi.org/10.1007/s10489-011-0287-y

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE : Synthetic Minority Over-sampling Technique. 16*, 321–357.

Chvatal, V. (1979). A Greedy Heuristic for the Set-Covering Problem. In *Source: Mathematics of Operations Research* (Vol. 4, Issue 3). https://www.jstor.org/stable/3689577.

[dataset] Andrea, Dal Pozzolo. (2017) *Credit Card Fraud Detection.*. Https://Www.Kaggle.Com/Datasets/Mlg-Ulb/Creditcardfraud. *Accessed March 27, 2024.*

Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research, 61*, 863–905. https://doi.org/10.1613/jair.1.11192

Gao, X., Ren, B., Zhang, H., Sun, B., Li, J., Xu, J., He, Y., & Li, K. (2020). An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling. *Expert Systems with Applications, 160*. https://doi.org/10.1016/j.eswa.2020.113660

Garey, M. R., & Johnson, D. S. (1979). *Garey, David S. Johnson - Computers and Intractability - A Guide to the Theory of NP-Completeness* (1st ed.). https://doi.org/10.1090/S0273-0979-1980-14848-X.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., & Yuanyue, H. (2017). Learning from class-imbalanced data : Review of methods and applications. *Expert Systems With Applications, 73*, 220–239. https://doi.org/10.1016/j.eswa.2016.12.035

Johnson, D. S. (1974). Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences, 9*.

Knowles, B., D'Cruz, J., Richards, J. T., & Varshney, K. R. (2023). Humble AI. *Commun. ACM, 66*(9), 73–79. https://doi.org/10.1145/3587035

Kou, Y., Lu, C., & Sinvongwattana, S. (2004). *Survey of Fraud Detection Techniques Yo-Ping Huang*. 749–754. https://doi.org/10.1109/ICNSC.2004.1297040.

Lebichot, B., Paldino, G. M., Siblini, W., He-Guelton, L., Oblé, F., & Bontempi, G. (2021). Incremental learning strategies for credit cards fraud detection. *International Journal of Data Science and Analytics, 12*(2), 165–174. https://doi.org/10.1007/s41060-021-00258-0

Liu, X. Y., Wu, J., & Zhou, Z. H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39*(2), 539–550. https://doi.org/10.1109/TSMCB.2008.2007853

Lovász, L. (1975). On the ratio of optimal integral and fractional covers. *Discrete Mathematics, 13*.

Lunardon, N., Menardi, G., & Torelli, N. (2014). ROSE: A package for binary imbalanced learning. *R Journal, 6*(1), 79–89. https://doi.org/10.32614/rj-2014-008.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology, 5*(9), 1315–1316. https://doi.org/10.1097/JTO.0b013e3181ec173d

Peeperkorn, J., vanden Broucke, S., & De Weerdt, J. (2024). Validation set sampling strategies for predictive process monitoring. *Information Systems, 121*. https://doi.org/10.1016/j.is.2023.102330.

González-Fabra, J., Álvarez-Moreno, M., Gumbau, M., & Bo, C. PubChem. (2017, July 12). *Bioassay Datasets*. Https://Www.Kaggle.Com/Datasets/Uciml/Bioassay-Datasets. https://doi.org/https://doi.org/10.19061/iochem-bd-6-3, *Accessed March 27, 2024*.

Vasconcelos, M. O., & Cavique, L. (2022). Dataset for corruption risk assessment in a public administration. *Data in Brief, 40*, Article 107768. https://doi.org/10.1016/j.dib.2021.107768

William, W., Ware, A., Basaza-Ejiri, A. H., & Obungoloch, J. (2018). A review of image analysis and machine learning techniques for automated cervical cancer screening from pap-smear images. *Computer Methods and Programs in Biomedicine, 164*, 15–22. https://doi.org/10.1016/j.cmpb.2018.05.034

Wolsey, L. A. (2021). *Integer programming* (John Wiley & Sons, Ed.; 2[a]). WILEY. https://doi.org/DOI:10.1002/9781119606475.

Zhu, B., Baesens, B., Backiel, A., & Vanden Broucke, S. K. L. M. (2018). Benchmarking sampling techniques for imbalance learning in churn prediction. *Journal of the Operational Research Society, 69*(1), 49–65. https://doi.org/10.1057/s41274-016-0176-1