Nogales *et al. Journal of Big Data* (2025) 12:141
https://doi.org/10.1186/s40537-025-01188-5

Journal of Big Data

METHODOLOGY Open Access

Self-organizing maps to evaluate optimal strategies for balancing binary class distributions: a methodological approach

Alberto Nogales[1*], Diego Guadalupe[1] and Álvaro J. García-Tejedor[1]

*Correspondence:
alberto.nogales@ceiec.es

[1] CEIEC, Research Institute, Universidad Francisco de Vitoria, Ctra. M-515 Pozuelo-Majadahonda Km 1800, 28223 Pozuelo de Alarcón, Spain

Abstract

Since machine learning algorithms rely on data, the way datasets are collected significantly impacts their performance. Data must be carefully gathered to minimize missing values or class imbalance. However, the inherent nature of the data tends can sometimes lead to such imbalances. An unbalanced dataset can lead to biased models, where predictions are influenced by the majority class. To avoid this problem, balancing strategies can be applied to equalize the instances of each class. This paper introduces a methodological approach to evaluate which balancing strategies yield the best results depending on the dataset. We leverage self-organizing maps, an unsupervised neural network model, to identify which strategy generates the most suitable balanced synthetic data. By considering the topological structure of the data, we propose a metric that uses the trained map to measure changes between the original dataset and the transformed dataset after applying different strategies. This metric is based on the idea that synthetic data resembling the original dataset more closely is preferable.

Keywords: Unbalanced datasets, Balancing strategies, Artificial intelligence, Machine learning, Self-organizing map

Introduction

The performance of machine learning (ML) models is determined by the quantity and quality of the data used for training. While data availability increases annually, the quality does not necessarily improve at the same rate. It is essential to curate data for these models, transforming raw data into a format and quality that is usable by the algorithms. This process accounts for 70% of the whole ML pipeline [1].

This step requires standardized data collection strategies and careful quality control, which are often not adequately met. This problem leads to datasets with missing values, differences in data strings, or imbalanced class features. The latter introduces a learning bias towards the majority class that can be avoided by applying balancing strategies [2].

There are two possible causes for datasets being imbalanced: intrinsic or extrinsic factors [3]. The former is due to the nature of the instances; for example, when collecting data for cancer diagnosis, most medical tests correspond to healthy individuals. The latter occurs

during the collection process due to the lack of standard methods, storage problems, or similar situations.

A commonly used method for handling highly imbalanced datasets is resampling. This technique involves either reducing the number of samples in the majority class (undersampling) and/or increasing the number of samples in the minority class (oversampling) by generating synthetic data. Typically, both types of strategies are combined in what are known as hybrid balancing strategies. Given the high number of undersampling and oversampling strategies available, selecting the most effective method for a specific problem can be a complex and time-consuming task. The imbalance in class distribution within datasets poses a significant challenge, often leading to biased models and poor predictive performance. Therefore, it is crucial to establish a reliable evaluation method to determine which combination of techniques yields the best results. The aim of this study is to address this need by proposing a robust strategy to facilitate the decision-making process. In this strategy, we recognize that synthetic data are more effective when they are more similar to the original data. This is particularly important in fields such as medicine, where generating data plays a role in decision-making.

The main contribution of this study is a methodological approach that employs an unsupervised neural network model known as self-organizing maps (SOMs) or Kohonen maps to systematically evaluate various combinations of undersampling and oversampling strategies. In addition, a new metric is introduced to achieve the best balance and performance for a given dataset. SOMs are based on biological studies of the cerebral cortex and were introduced in 1982 by Kohonen [4, 5]. They are artificial neural networks with a nonsupervised training algorithm that is particularly effective for visualizing high-dimensional data, performing nonlinear mapping between high-dimensional patterns and a discrete bidimensional representation, called a feature map, without external guidelines. For this reason, SOMs have been widely used as a method of pattern recognition, dimensionality reduction, data visualization, and, especially, clustering since unsupervised training guarantees bias-free results. The innovation of this study lies not in the use of existing methods such as balancing strategies or SOMs but in the design of a novel approach that integrates these techniques into a comprehensive pipeline designed to assess and optimize balancing strategies on the basis of the specific characteristics of different datasets. Additionally, this paper introduces a unique metric, a key contribution that leverages the topological properties of SOMs to evaluate and compare the effectiveness of different balancing methods. This metric provides a new perspective for selecting the most suitable balancing strategy, addressing limitations in existing evaluation techniques.

The remainder of the paper is structured as follows. Sect."Related works"provides a summary of related works with similar approaches. Sect."Materials and methods"details the datasets and methods used during the study. Sect."Results"presents the results of the evaluation and discussion. Sect."Conclusions and future work"provides conclusions and directions for future work.

Related works

In this paper, we compare different balancing strategies using Kohonen maps as a method to evaluate the effectiveness of synthetic data related to that of the original data. Similarly, we review papers proposing new approaches for handling imbalanced datasets

and others that are aimed at evaluating the generation of synthetic data with different strategies.

In the first category of studies, the following works were identified. In Chawla et al. [6], two methods were introduced, a new version of synthetic minority over-sampling technique (SMOTE) and a novel approach named SMOTEBoost. These methods were evaluated using the AUC, precision, and recall. A strategy using the neighbour-cleaning rule (NCR) and SMOTE was applied [7] to imbalance medical data and evaluated via K-nearest neighbour (KNN), sequential minimal optimization (SMO), and naïve Bayes (NB). Another hybrid method was presented [8]; in this case, NCL was employed for oversampling, adaptive semisupervised weighted oversampling (ASUWO) was applied for undersampling, and the results were evaluated with a decision tree (DT) and random forest (RF).

Works aimed at evaluating imbalanced strategies are summarized below. Wainer and Franceschinell [9] evaluated 20 strategies, such as Tomek links (TL) or one-sided selection (OSS), over a total of 58 datasets via a support vector machine (SVM) with a radial basis function (RBF) kernel, RF, and gradient boosting machines using six different metrics. Their findings suggest that each strategy's effectiveness varies considerably depending on the metric applied. Another evaluation was reported in Costa et al. [10], where a meta-learning approach was applied to evaluate nine imbalanced strategies using the edited nearest neighbour (ENN) or SMOTE and tested using 163 datasets via SVM. The authors concluded that the most suitable strategy depends on the features of the dataset. For example, SMOTE-TL is more suitable for more challenging classification tasks and high-dimensional datasets. SVM was employed to evaluate ten imbalanced strategies for the task of text classification in three benchmarks [11]. The authors identified SMOTE as the best resampling method for imbalanced text. Although SMOTE performs slightly better in some cases, the differences are minor and inconsistent across datasets. Overall, optimal thresholding has a greater influence on the performance of balancing strategies. Additionally, Goel et al. [12] evaluated five different strategies on the basis of five metrics via SVM. In this case, the conclusions indicate that, depending on the performance metric, the best sampling method changes. Then, Shamsudin et al. [13] evaluated the combinations of the random undersampling strategy (RUS) with SMOTE, ADASYN, borderline, SVM-SMOTE, and the random oversampling strategy (ROS) using a DT. The results were compared with those from the literature, which shows that hybrid strategies are better than simpler strategies are and that the problem is that the study is only conducted with one dataset. A different evaluation was performed in Gosain and Sardana [14], where the oversampling strategies such as SMOTE, BSMOTE, ADASYN, and SLSMOTE were applied to seven datasets and evaluated with SVM, KNN, and NB. In this case, SLSMOTE outperforms the other methods. However, depending on the dataset and the metric, other strategies can perform better. Another interesting work is that by Kraiem et al. [15], who examined the effectiveness of seven resampling methods (condensed nearest neighbour (CNN), among others) to address class imbalance in 40 datasets. The authors analysed how data characteristics, such as the imbalanced ratio, sample size, number of attributes, and class overlap, impact the performance of these resampling strategies in improving classification outcomes via RF. The findings indicate that SMOTE-based methods generally yield better results,

particularly in highly-imbalanced datasets. In the Wongvorachan et al. [16], three resampling methods were compared using an educational dataset via the RF classifier. The results indicated that the ROS method performed best for moderately imbalanced data, whereas the hybrid method excelled with extremely imbalanced data. In Mujahid et al. [17], an evaluation of five oversampling techniques was performed. Two highly imbalanced Twitter datasets were selected, and the performance of these methods was compared across six classifiers. The results indicate that ADASYN and SMOTE achieved the best accuracy and recall, particularly with SVM, but no single method universally outperformed the others across all the models and metrics. Alamri and Ykhlef [18] proposed a hybrid approach that combines Tomek links, BIRCH clustering, and borderline SMOTE (BCBSMOTE) to handle imbalanced credit card data. Improved performance was achieved when employing a random forest in terms of F1-score, AUC-ROC, and other metrics. Yang et al. 2024 [21] investigated the impact of ROS and RUS on clinical prediction models via Lasso logistic regression, RF, and XGBoost. They reported that these methods generally do not enhance performance metrics and did not recommend the use of these strategies to generate synthetic data. Parrales-Bravo et al. [19] examined the effects of various oversampling and undersampling techniques on NB classifiers for predicting preeclampsia and concluded that while these techniques improve sensitivity and specificity, they do not guarantee better accuracy (synthetic data do not always lead to improved model performance). Finally, Santoso et al. [20] reviewed synthetic oversampling methods for addressing imbalanced data, emphasizing that each method generates unique synthetic data characteristics and must be chosen on the basis of specific imbalance levels and patterns. The authors concluded that no single method is universally effective for managing class imbalance. In Table 1, we summarize the main features of the previous papers.

As observed, while other evaluations exist, to our knowledge, this is the first study using SOMs by leveraging their topological properties and introducing a metric on the basis of feature characteristics. We also note that almost all the previous works suggest that regardless of the evaluation method, the best-performing strategy depends on the data and the evaluation metrics used. With respect to the use of SOMs, we have identified studies employing this model to improve robustness to noise and adapt to dynamic data streams by benefiting from their topological structure. Yu et al. [22] proposed a Gaussian membership-based self-organizing incremental neural network (Gm-SOINN), which addresses the stability–plasticity problem in traditional SOINNs by integrating fuzzy logic through Gaussian memberships. Similarly, a topology learning-based fuzzy random neural network (TLFRNN) was introduced to handle streaming data regression with a focus on noise reduction and adaptability to concept drift through multiple fuzzy sets and a randomness layer [23].

In the present work, the use of Kohonen maps facilitates the construction of a topological map, enabling the development of a metric to measure the performance of different hybrid strategies. The metric measures how similar synthetic data are to the original data. This approach makes our method unique, as the evaluation of the strategies is based on a comparison with the original data and not on how the balanced dataset performs on an ML model, as in previous works. In this sense, our method rewards the generation of synthetic data that more closely resemble the original data. This characteristic

Table 1 Summary of state-of-the-art sampling methods

References	Sampling Methods	Metrics	Classification Models
[6]	SMOTE, SMOTEBoost, AdaCost	Precision, Recall, F1-score	RIPPER
[7]	NCR, SMOTE	Recall	KNN, SMO, NB
[8]	NCR, A-SUWO	Accuracy, Recall, Precision, F1-score	C4.5, RF
[9]	20 strategies (TL, OSS, ADASYN, etc.)	Accuracy, Precision, Recall, Specificity, F1-score, AUC, Balanced Accuracy, G-Mean, MCC	SVM, RF, GB
[10]	9 strategies (SMOTE, ENN, ADASYN, etc.)	F1-score	SVM
[11]	SRAND, CLUS, SMOTE	Recall	SVM
[12]	SMOTE, ADASYN BorderlineSMOTE, SMOTE-Tomek, RUSBoost	Accuracy, Precision, Recall, F1-score, G-Mean	SVM
[13]	RUS, SMOTE, ADASYN, SLSMOTE	Precision, Recall, F1-score	DT
[14]	SMOTE, BSMOTE, ADASYN, SLSMOTE	Accuracy, Specificity, Precision, F1-value, G-Mean, AUC	SVM, KNN, NB
[15]	OSS, TL, ENN, CNN, SMOTE	Accuracy, Precision, F1-score, Recall, AUC, G-Mean, Optimal Precision, Index of Balanced Accuracy	RF
[16]	ROS, RUS, SMOTE-NC + RUS	Accuracy, Precision, Recall, AUC, F1-score	RF
[17]	SMOTE, SVM-SMOTE, BorderlineSMOTE, K-Means SMOTE, ADASYN	Accuracy, Recall	RF, SVM, AdaBoost, RL, DT
[18]	TL, BCBSMOTE	F1-score, Accuracy, AUC-ROC, Precision, Recall	RF
[21]	ROS, RUS	AUC, F1-score	Lasso Logistic Regression, RF, XGBoost
[19]	SMOTE-NC, SMOTE-ENC, ROSE, ROS, RUS	Accuracy, Sensitivity, Specificity, F1-score, AUC	NB,
[20]	5 SMOTE versions	Accuracy, Precision, Recall, F1-score	Not specified

is important in fields such as medicine, where maintaining similar to real data is essential due to the sensitivity of applications. We should also emphasize that this approach aims to evaluate the most effective strategy for different studied datasets. As demonstrated in related works, the optimal strategy often varies depending on different issues, such as the dataset or applied metric. Consequently, it is challenging for studies of this nature to categorically conclude that any single strategy is universally superior. Therefore, we believe that the different state-of-the-art approaches provide different perspectives that are all valid and that the users should decide which of them adapts better to the use case.

Materials and methods

This study aims to develop a step-by-step methodology based on SOMs to identify the most suitable balancing strategy for a given use case. The proposed workflow involves selecting an unbalanced dataset and applying a combination of oversampling and undersampling techniques. A Kohonen map is subsequently trained using the original dataset, where the synthetic data generated in the first step are classified in the next step. Although SOMs are typically not used as classifiers, previous studies [24] have demonstrated their potential for this purpose. If the distribution of the synthetic data closely

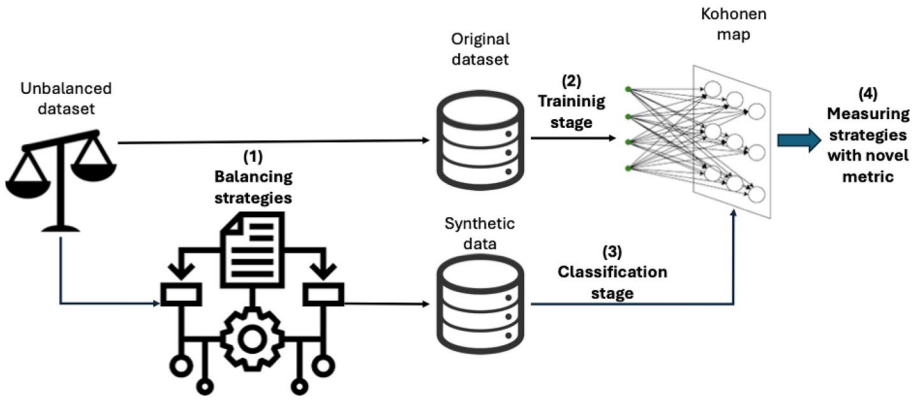

Fig. 1 Workflow of the proposed method

resembles that of the original data, classification errors on the trained map should be minimal. To quantitatively assess the performance of the balancing strategies, we introduce a novel metric that is based on the topological properties of the SOMs that are used in the last step. Figure 1 shows the implementation of this workflow.

Apart from this workflow, we employed multilayer perceptrons (MLPs) to verify that the dataset was accurately balanced for effective binary classification. This was confirmed by analysing accuracy metrics, which indicated that the models achieved an appropriate bias–variance trade-off. The results should demonstrate consistent performance across testing and minimal differences between the training, validation, and test stages, suggesting that overfitting was not an issue.

Unbalanced datasets

The datasets used in this study are described below. In total, we used 6 datasets: cancer breast, oil spill, German credit, phonemes, microcalcifications, and credit card fraud.

First, we chose the Haberman dataset for breast classification.[1] This dataset was compiled by the University of Chicago's Billings Hospital from 1958 to 1970. It was comprised of a binary classification for patients who died within 5 years or survived 5 years or longer. This classification includes 3 numerical features: patient age, year of surgery, and the number of positive axillary nodes detected. In total, the dataset contains 307 instances.

The second dataset is derived from oil spills in satellite radar images.[2] This dataset was introduced by Kubat et al. [25] and consists of satellite images of the ocean, some of which contain oil spills. Images were preprocessed, yield a set of 49 features that describe the images: area, intensity, or sharpness. The total number of images is 937.

The third case is referred to as the German credit dataset.[3] This dataset comprises a set of clients and some financial and banking features to predict whether a client will pay a loan back. This prediction is based on 7 integers and 13 categorical variables. These features include the duration in months, amount, present residence, and job. The information of 1,000 clients was compiled.

[1] https://archive.ics.uci.edu/ml/datasets/haberman's+survival

[2] https://www.kaggle.com/datasets/ashrafkhan94/oil-spill

[3] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

Table 2 Dataset summary

Dataset	Number of features	Number of missing values	Classification type	Imbalance
Breast cancer	3	0	Binary numerical	225/81
Oil spills	49	0	Binary numerical	896/41
German credits	20	0	Binary numerical	700/300
Phonemes	5	0	Binary numerical	3,818/1,586
Microcalcifications	6	0	Binary numerical	10,923/260
Credit card fraud	6	0	Binary numerical	284,315/492

Fourth, is the phonemes dataset.[4] This dataset is aimed at distinguishing between nasal and oral sounds. This is achieved using a set of 5 features that characterize the amplitude of the first five harmonics normalized by the total energy. In total, 5,427 examples were compiled.

As a fifth use case, we chose the microcalcification dataset.[5] This dataset is used for breast cancer detection from radiological scans. Specifically, it focuses on identifying clusters of microcalcification, which appear bright on mammograms. The dataset was curated by scanning the images, segmenting them into candidate objects, and employing computer vision techniques to characterize each candidate object by using six features.

Finally, we selected the credit card fraud detection dataset.[6] The dataset comprises transactions conducted by European cardholders using credit cards in September 2013. It only includes numerical input variables resulting from a principal component analysis (PCA) transformation. Among the 30 features, 28 are principal components derived from PCA, while the remaining 2 were not transformed by PCA. The details of all the datasets are summarized in Table 2.

Balancing strategies

In this paper, we evaluate strategies to avoid the problem of unbalanced classes in several datasets. There are many types of imbalanced strategies. However, we adopted hybrid strategies, as they have been shown to yield better data distributions and improve performance in classification problems [26]. In hybrid strategies, synthetic data are generated from the minority class. Then, instances are removed from both distributions. This approach not only mitigates the class imbalance problem but also eliminates noisy instances that may have been incorrectly positioned on the opposite side of the cluster boundary. Next, we define all the over- and undersampling strategies that we incorporated into the hybrid strategies we are evaluating in the paper. We chose to use these sampling methods as a first step to test our method with well-established or classical techniques as a foundational approach. In the case of oversampling methods, Chen et al. [27] highlighted that most of them are frequently used in practice. For undersampling methods, some papers [28] or [29] identified these methods as basic methods. By referring to Table 1, we can also confirm that most of the following strategies have been evaluated in previous works.

[4] https://datahub.io/machine-learning/phoneme

[5] https://www.kaggle.com/datasets/sudhanshu2198/microcalcification-classification

[6] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Oversampling strategies

This type of approach for handling imbalanced data generates synthetic instances of the minority class to balance the number of instances per class.

Synthetic minority oversampling technique (SMOTE) This method, which was originally introduced by Chawla et al. [30], generates data instances without replacing the original data. SMOTE selects instances of a feature space, establishing a line between them. Then, a new instance is generated at a randomly chosen point along the line.

Adaptive synthetic sampling (ADASYN) This method, which was presented by He et al. [31], uses a density distribution to automatically determine the number of synthetic samples required for each minority data instance. This density distribution serves as a measure of the weight distribution among various minority class examples, reflecting their respective learning difficulties. Consequently, the application of this method not only ensures a balanced representation of the data distribution at the desired balance level but also focuses the learning algorithm's attention toward challenging instances.

Borderline SMOTE As described by Han et al. [32], this technique involves identifying borderline instances within the minority class. These borderline instances are then utilized to generate new synthetic examples. These synthesized instances are strategically positioned around the borderline examples of the minority class.

SVM SMOTE Nguyen et al. [33] proposed this method, which includes the following three stages. First, the minority class is oversampled to effectively address data imbalance. Second, the sampling strategy is focused primarily on critical regions, particularly the boundary area between classes. Third, extrapolation is applied to extend the minority class region, especially in areas where the majority class instances are scarce.

K-Means SMOTE The approach presented in Last et al. [34] involves three main steps: clustering, filtering, and oversampling. In the clustering step, the input space is divided into k groups via K-means. Next, in the filtering step, clusters with a significant proportion of minority class samples are retained for oversampling. The number of synthetic samples to generate is subsequently distributed, with more samples assigned to clusters containing sparsely distributed minority samples. Finally, in the oversampling step, SMOTE is applied within each selected cluster to achieve the desired ratio between minority and majority instances.

Undersampling strategies

These strategies involve reducing the number of instances from the majority class to balance the number of instances across classes.

Tomek links (TL) As described by Tomek [35], this technique removes boundary instances that are more likely to be misclassified. This method is based on the definition of a Tomek-link pair, which occurs when two instances belong to the different classes and have no other example with a smaller distance to the first instance. In summary, if

instances form a Tomek-link pair, there are more possibilities of having superfluous data along the distribution.

Edited nearest neighbour (ENN) Introduced by Wilson [36], this method aims to refine datasets by removing samples from the majority class that lie close to the decision boundary. If the label of a majority class instance and the labels of applying K-nearest neighbours differ, then both the instance and its nearest neighbours are removed from the dataset.

Condensed nearest neighbour rule (CNNR) This was the first selection algorithm [37]. It employs two storage areas: the condensed set (CS) and the training set (TS). Initially, the TS includes the complete training set, where the CS remains empty. To initiate the process, an instance is randomly selected from the TS and moved to the CS. Each instance $x \in$ TS is subsequently compared with those currently stored in the CS.

Neighbourhood cleaning rule (NCL) This algorithm [38] has two stages. The process begins with the application of the edited nearest neighbour algorithm to undersample instances that do not belong to the target class. In the second step, the neighbourhood of the remaining examples is refined. Here, the KNN algorithm is applied, removing an example if its neighbours do not belong to the target class and if the example's class exceeds half of the smallest class within the target class.

One-sided selection (OSS) As described by Kubat and Matwin [39], this method reduces the number of misclassified instances by creating a subset with the training set. Following this, the method removes misclassified examples involving Tomek links. This process discards noisy and borderline examples, resulting in the formation of a new training set.

Self-organizing maps

The SOM, which in this work is referred to as a Kohonen map, establishes a relation between a higher-dimensional input space and a lower-dimensional map space via a two-layered fully connected architecture. The input layer comprises a linear array with the same number of neurons as the dimension of the input data vector (n). The output layer, known as the Kohonen layer, consists of neurons, each with an associated weight vector of the same dimension as the input data (n) and a position in a rectangular grid of arbitrary size (k). These weight vectors are organized in an n * k * k matrix known as a weight matrix. Self-organization implies that a vector from the input dataset space (X) is presented to the network, and the node with the closest weight vector Wj is identified as the winning neuron or best matching unit (BMU) via a simple discriminant function (Euclidean distance) and a'winner-takes-all'mechanism (competition). The unsupervised training algorithm subsequently adjusts the winner's weight vector on the basis of its similarity to the input vector. This presentation of vectors from the input space and BMU learning continues until a specified number of presentations (P) is reached or the values of the selected metrics remain steady. The iterative process yields a trained (self-organized) Kohonen map, represented by a given weight matrix. Each node in the Kohonen layer corresponds to a specific pattern learned during training and can recognize all the elements belonging to that class. The self-organizing training process preserves the topological properties of the input space, allowing neighbouring nodes to

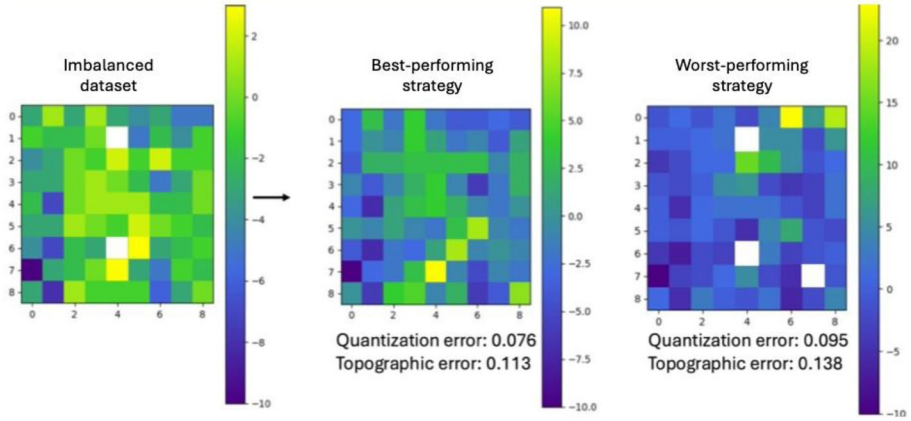

Fig. 2 Three heatmap for an SOM: imbalanced dataset (left), data after applying the best-performing strategy (middle) and data after applying the worst-performing strategy (right)

recognize patterns that are closer in the n-dimensional space, meaning that they have similar characteristics. The map generated by this trained SOM can then be used to classify additional input data through a process called"mapping."Unlike training, this process does not alter the weight matrix. New elements from the input space are placed where they are recognized by an existing best matching unit (BMU), indicating that they are similar (belonging to the same class) to those previously recognized by that BMU.

Similarity over maps activation-based grid topology (SOM-AGT) index

One of the contributions of this work is the proposal of a new metric using the concept of topology provided by Kohonen maps. The idea is that the neurons in the map trained with the unbalanced dataset should recognize similar input vectors when the synthetic data are mapped. Therefore, topological changes from one map to another should be small. For this purpose, we use the instances of the map that we consider the best trained, which are those with the lowest errors.

To validate this assumption, we use two derived graphical representations of Kohonen maps. The first representation is a heatmap that indicates differences between the number of synthetic instances recognized by the neurons that belong to each class. Neurons recognizing more instances of the minority class (oversampled instances) are shown in yellow, where those recognizing more instances of the majority class appear in blue. Empty neurons are blank. In Fig. 2, we show an example of this type of map using the breast cancer dataset. The figure shows, from left to right, the heatmap generated from the imbalanced dataset, the map after using the best imbalanced data, and the map using the balanced data of the strategy that performs the worst.

The second map uses four colours to represent how neurons recognize instances of the two classes in the datasets. Next, we explain the four use cases and their meanings:

- Blank or empty neurons indicate that no instances are recognized.
- Red neurons recognize a greater percentage of instances from the minority class. The percentage acts as a threshold that can be varied to perform different analyses, indi-

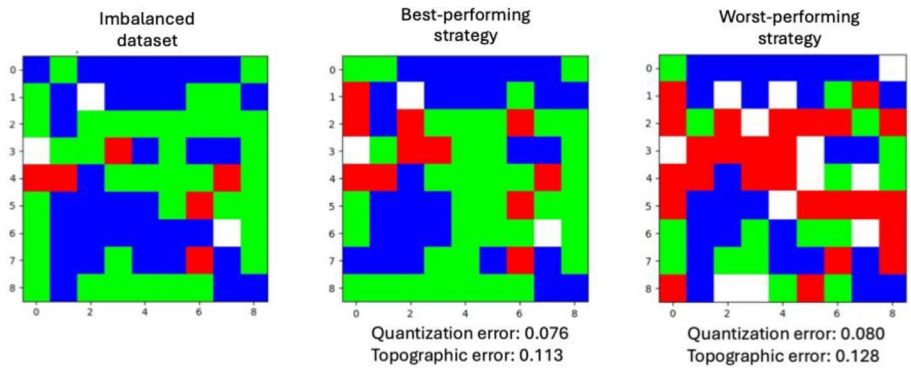

Fig. 3 The pure colour map shows three versions: imbalanced dataset (left), with the best-performing strategy (middle) and the worst-performing strategy (right)

cating the percentage of instances from one of the two classes (minority or majority) that must be exceeded to colour the neuron with the corresponding class colour.

- The blue neurons are the same as the red neurons but for the majority class.
- Green neurons represent a balance in recognizing instances of both classes without either exceeding the set threshold.

Figure 3 shows the same information as that in Fig. 2 but employs pure colours for the representation. The map on the left side contains the imbalanced dataset, the map in the middle represents the dataset after the strategies that perform the best are applied, and the map on the right side contains the map after the worst-performing strategy is applied.

These maps support the idea that the strategies that yield lower errors are those that generate synthetic data closely resembling the original class instances. This suggests that successful strategies generate synthetic samples that maintain the characteristics and distribution of the original data, thereby improving model performance. As shown in both figures, the changes from the first map to the second map are lower, which indicates that the instances generated through the balancing strategies better align with the original dataset. For example, if a red or blue neuron (indicating recognition of many instances of one class) changes to the opposite colour, we can infer that the synthetic data are of inferior quality. This is because the model confuses synthetic data to such an extent that a neuron, which should primarily detect one class, is now detecting many instances of the opposite class. Conversely, if a green neuron, which is on the borderline of being pure, turns red or blue, there is no immediate issue. This outcome simply means that the neuron has recognized one additional instance of one of the classes, making it a pure neuron. Similarly, if a red or blue neuron becomes green, it has recognized one more instance of the opposite class, making it nonpure, which is also not problematic. At this point, we have demonstrated that for a given use case, when an SOM trained with the unbalanced dataset classifies data generated by the best balancing strategies (those that produce maps with the lowest quantization and topological errors), the mapping process exhibits only slight changes. This finding indicates that the best balancing strategies generate synthetic data with a distribution that closely matches

that of the original data, maintaining the integrity and effectiveness of the SOM's classification capabilities.

However, these metrics show minimal differences. Therefore, we need to establish a method to measure the validity of the different balancing strategies. For this purpose, we propose our metric on the basis of the idea described above. The similarity is based on the Jaccard index [40] and defined in the following equation.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{1}$$

where A and B are two different sets, |A ∩B| is the number of elements in the intersection of sets A and B, and |A ∪B| is the number of elements in the union of both sets. The Jaccard index ranges from 0 to 1, where 0 indicates that the two sets are disjoint (no common elements) and where 1 indicates that the two sets are identical.

We have adapted this index to the graphical representation of Kohonen maps, which we refer to as the similarity over maps activation-based grid topology (SOM-AGT) index. The two sets correspond to the mapping of the original dataset (KM_1) and the mapping after applying balancing strategies (KM_2). To compute the index, the Jaccard index is applied to red-, blue-, and green-coloured neurons. Then, we average the values, providing a percentage of similarity. This metric is formalized as follows.

$$SOM - AGT(KM_1, KM_2) = \frac{J(KM_1, KM_2)_{RED} + J(KM_1, KM_2)_{BLUE} + J(KM_1, KM_2)_{GREEN}}{3} \tag{2}$$

To establish this metric, we need to demonstrate that it satisfies the following properties, as outlined in Salabun and Shekhovtsov [41]: nonnegativity (distances are always zero or positive), identity of indiscernibles (the distance between an object and itself is zero), symmetry (the distance from A to B is equal to the distance from B to A), and triangle inequality (the direct distance from A to B is always less than or equal to the distance from A to B via an intermediate point C).

For nonnegativity, as $J(A,B)$ ranges from 0 to 1, it follows that for any colour, $1-$ $.SOM\text{-}AGT\,(KM_1, KM_2) \geq 0$.

In the case of the identity of indiscernibles, if $SOM\text{-}AGT\ (KM_1, KM_2) = 1$, then $KM_1 = KM_2$ with a value of 1 for any colour because $|KM_1 \bigcap KM_2|$ is equal to the denominator $|KM_1 \bigcup KM_2|$ for any colour.

With respect to symmetry, as the Jaccard index is inherently symmetric $d(A,B) = 1 - J(A,B) = 1 - J(B,A) = d(B,A)$, this condition holds for $SOM\text{-}AGT$ (KM_1, KM_2).

Finally, to demonstrate triangle inequality for any three mappings KM_1, KM_2 and KM_3: $d(KM_1, KM_3) \leq d(KM_1, KM_2) + d(KM_2, KM_3)$. Given that $d(KM_i, KM_j) = 1 - SOM\text{-}AGT(KM_i, KM_j)$ and considering that $SOM\text{-}AGT(KM_i, KM_j)$ is based on the average of Jaccard similarities for each colour, we leverage the fact that the Jaccard index itself satisfies the triangle inequality.

Table 3 Grid search values to train Kohonen maps

Hyperparameter	Values
Side map	[5–25]
Epochs	500, 1000, 2500, 5000, 7500, 10,000
Learning rate	0.01, 0.05, 0.1, 0.2, 0.3

Multilayer perceptron (MLP)

This model consists of sequential layers composed of neurons, with each layer connected to adjacent layers. It requires a minimum of three layers: input, hidden, and output. Input data are introduced through the input layer, processed in the hidden layer, and classified by the output layer. MLPs optimize parameters through a two-stage backpropagation training process: forwards and backwards, as described by Rumelhart et al. [42].

Results

Next, we present all the results obtained during the application of our proposed approach. The results are organized step by step, incorporating values that provide insights during the process and serve as support material for decision-making.

First, we need to apply all the strategies to the unbalanced datasets, leading to a total of 25 combinations (5 oversampling strategies and 5 undersampling strategies). Then, for each of these combinations, we train a Kohonen map. For this purpose, we use the GEMA Python library developed by García-Tejedor and Nogales [43]. All the maps are trained via a grid search strategy, which identifies the optimal value of the hyperparameters by aggregating various ranges of possibilities [44]. To avoid problems caused by random weight initialization, each neuron in the Kohonen layer is weighted from one of the input instances. On the basis of the main function of the SOM, which, according to Khalilia and Popescu [45], is topology preservation of the input data, the overall topology tends to remain consistent across instances. In Table 3, we compile all the hyperparameters and values used for this stage.

To find the optimal Kohonen map, we use quantization and topographic errors. The quantization error represents the mean distance between each data vector and its BMU. This metric is calculated for the winning neurons and is independent of the number of"empty"neurons and the size of the map, serving as a measure of map resolution. This error is defined in Eq. 3.

$$QE = \frac{1}{N}\sum_{i=1}^{N} \|X_i - BMU_{(i)}\| \tag{3}$$

As denoted above, N is the number of instances in the training datasets, and X_i is an input vector.

Moreover, the topographic error indicates the ratio of all the data vectors for which the first and second BMUs are not adjacent units, providing insight into topology preservation. Equation 4 defines the topographic error.

$$TE = \frac{1}{N}\sum_{i=1}^{N} t(x_i) \tag{4}$$

where $t(x_i)$ equals 0 if the BMU and the second-best matching units are adjacent; otherwise, its value is 1, and N is the total number of instances.

A sensitivity analysis of the grid search for each dataset is provided in the Figures compiled in Appendix A. This information provides deeper insights into model behaviour, ensuring robust, interpretable, and efficient model tuning.

In Table 4, for each selected dataset, both metrics are used for all the combinations of imbalanced strategies applied to each dataset via GEMA. All the raw data related to these experiments can be found in Appendix B.

In the Table 4, the top five strategies are compiled for each dataset. These strategies are identified the most effective, as they yield lower values for both quantization and topological errors. In addition, the best strategy, appearing in the first position, is highlighted in bold.

For the bank loan dataset, the top five strategies include SMOTE combined with the ENN, CNN, and NCR and ADASYN with the CNN and NCR. For the phonemes dataset, ADASYN with the CNN and TL are identified among the top strategies. The combination of Borderline SMOTE and the TL and CNN are the other highlighted strategies. With respect to the breast cancer dataset, SMOTE with the ENN and OSS, ADASYN with the CNN and OSS, and K-means SMOTE with the OSS are among the best-performing strategies. For the credit fraud dataset, the leading strategies are as follows: ADASYN combined with the TL, ENN, and NCR and K-means SMOTE with the OSS. For the oil spill dataset, K-means SMOTE with the ENN is at the top-performing strategy. Other top-performing strategies include K-means SMOTE with the CNN, NCR, and OSS and ADASYN with the NCR. Finally, for the microcalcification dataset, SVM-SMOTE combined with five other strategies performed best.

The distribution of top-performing strategies varies, with only two of them appearing three times on the best: ADASYN with the NCR and K-means SMOTE with the OSS. Among the most frequently used sampling strategies, NCR appears 8 times, CNN 6 times, and ENN 6 times. In the case of the oversampling methods, ADASYN was used 10 times, K-means SMOTE was used 7 times, SMOTE was used 5 times, and SVM plus SMOTE was used 5 times.

As the differences in the error metrics between different strategies are incredibly low in most cases, we can conclude that the errors are intuitive but not conclusive. On this basis, we applied the proposed metric. Next, we present the values of our metric after applying the balancing strategies to the six proposed datasets. All of this information is compiled in Table 5, separated by dataset. The columns in the table indicate the percentages corresponding to the threshold that defines a neuron as pure (red or blue). For each dataset, the metric values are provided for the three strategies that yield the best results. For each threshold and strategy, we obtain the average value and standard deviation to determine whether there is stability among the results.

As can be seen, the threshold has minimal impact on all the cases. So, we have selected a threshold of 80% as it allows us to identify pure neurons more accurately. If we look at the strategies separately, we can conclude that the differences also are not

Table 4 Kohonen map errors for the datasets

		Quantization error	Topographic error
Bank loan dataset	**SMOTE + CNN**	**0.912**	**0.172**
	ADASYN + CNN	**0.912**	**0.172**
	SMOTE + ENN	0.916	0.174
	ADASYN + NCR	0.914	0.179
	SMOTE + NCR	0.914	0.179
Phoneme dataset	**ADASYN + ENN**	**0.141**	**0.207**
	Borderline SMOTE + NCR	0.141	0.208
	Borderline SMOTE + CNN	0.144	0.206
	Borderline SMOTE + TL	0.145	0.206
	ADASYN + TL	0.142	0.212
Cancer breast dataset	**ADASYN + CNN**	**0.080**	**0.086**
	SMOTE + OSS	0.076	0.105
	ADASYN + OSS	0.076	0.105
	K-Means SMOTE + OSS	0.079	0.103
	SMOTE + ENN	0.076	0.113
Credit fraud dataset	**K-Means SMOTE + OSS**	**0.196**	**0.152**
	K-Means SMOTE + NCR	0.239	0.112
	ADASYN + TL	0.221	0.156
	ADASYN + NCR	0.221	0.157
	ADASYN + ENN	0.221	0.158
Oil spill dataset	**K-Means SMOTE + ENN**	**0.513**	**0.146**
	K-Means SMOTE + NCR	0.525	0.133
	K-Means SMOTE + CNN	0.538	0.138
	K-Means SMOTE + OSS	0.538	0.145
	ADASYN + NCR	0.530	0.175
Microcalcification dataset	**SVM SMOTE + CNN**	**0.093**	**0.098**
	SVM SMOTE + TL	0.088	0.107
	SVM SMOTE + NCR	0.088	0.108
	SVM SMOTE + OSS	0.091	0.109
	SVM SMOTE + ENN	0.092	0.101

Bold values represent the best-performing strategy for each dataset

remarkably high, and they remain stable. The one considered as the best (first row for the dataset) does not stand out too much from the others but let us consider it as the best.

Now, as half of the datasets are unbalanced at around 40% and half are around 90%, we want to compare the performance of the strategies between datasets. Table 6 compiles the information related to the average and standard deviation of applying all the strategies. The results above show that the percentage of unbalanced data does not affect the quality of the synthetic dataset.

To establish an additional criterion for evaluating the effectiveness of the strategies, we analysed the frequency with which each strategy appears in the top three rankings across Table 5. This approach allows us to identify which strategies consistently perform well and are therefore more reliable in achieving optimal results. Table 7 presents the top-performing strategies.

Table 5 SOM-AGT coefficient for each dataset (n = 25)

Imbalanced strategy	Threshold 80%	Threshold 75%	Threshold 70%	Mean per strategy
Bank loan dataset (40% of unbalanced)				
KMSSMOTE + CNN	69.1	69.8	68.6	69.1 ± 0.54
SMOTE + ENN	69.7	67.3	68.1	68.3 ± 1.38
KMSSMOTE + OSS	66.5	65.4	65.7	65.8 ± 0.57
Mean total	63.5 ± 5.50	63.3 ± 3.30	62.8 ± 3.05	
Phoneme dataset (41% of unbalanced)				
SMOTE + ENN	76.7	76.1	74.9	75.9 ± 0.81
BSMOTE + CNN	74.5	72.1	73	73.8 ± 1.25
SMOTE + OSS	68.6	66.7	68	67.7 ± 0.92
Mean total	66.4 ± 5.49	63.1 ± 5.72	65.5 ± 6.00	
Breast cancer dataset (47% of unbalanced)				
SMOTE + ENN	81.1	71.4	71.9	74.8 ± 4.70
SMOTE + NCR	75.6	71.7	70.5	72.6 ± 2.81
BSMOTE + OSS	73.6	71.7	70.5	71.9 ± 1.61
Mean total	66.4 ± 6.15	63.8 ± 5.10	63.4 ± 5.30	
Credit fraud dataset (47% of unbalanced)				
KMSSMOTE + NCR	71.6	70.4	70.7	70.9 ± 0.61
ADASYN + ENN	67.4	66.5	66.8	66.9 ± 0.40
SMOTE + ENN	66.9	66.1	65.9	66.3 ± 0.41
Mean total	64.2 ± 4.70	63.8 ± 4.10	65.6 ± 3.80	
Oil spill dataset (91% of unbalanced)				
KMSSMOTE + NCR	79.2	73	73.3	75.1 ± 3.20
SMOTE + ENN	74.5	73.1	71.5	73.0 ± 1.46
KMSSMOTE + CNN	70.1	68.8	69	69.6 ± 0.68
Mean total	66.3 ± 6.20	64.9 ± 5.40	63.8 ± 5.30	
Microcalcification dataset (91% of unbalanced)				
ADASYN + TL	72.5	71.4	71.8	71.9 ± 0.55
KMSSMOTE + TL	70.1	69.4	69.2	69.5 ± 0.41
SVMSMOTE + TL	69.8	68.7	65.3	68.0 ± 2.46
Mean total	64.4 ± 5.40	63.8 ± 5.10	63.3 ± 5.80	

Table 6 Comparison of all strategies applied to the different datasets

Dataset (Unbalanced %)	Threshold = 80%
Bank loan (40)	63.54% ± 5.5
Phoneme (41)	66.4% ± 5.49
Breast cancer (47)	66.4% ± 6.15
Credit fraud (90)	64.2% ± 4.7
Oil spill (91)	66.3% ± 6.2
Microcalcification (91)	64.4% ± 5.4

Among these, only SMOTE + ENN stands out over the rest of the strategies. This finding aligns with the results obtained for the Kohonen map errors, where this SMOTE + ENN was considered one of the best strategies.

Once we trained an MLP via a grid search strategy for each of the datasets that yielded better performance, as shown in Table 4. On the basis of these results, we

Table 7 Top 5 strategies appearing in the top 3 rankings

Strategy	Frequency in top 3 rankings
SMOTE + ENN	5
KMSSMOTE + NCR	3
KMSSMOTE + CNN	2
SMOTE + OSS	2
SMOTE + NCR	2

Table 8 Trained MLPs after the best-balancing strategy is applied for each dataset

Dataset	Training %	Validation %	Test %
Bank loan	90.4% ± 1.8	86.2% ± 2.5	82.7
Phoneme	81.3% ± 5	80.6% ± 4.5	78
Breast cancer	97.3% ± 0.8	89.0% ± 6.0	87.3
Credit fraud	99.7% ± 0.2	99.6% ± 0.3	99.8
Oil spill	93.9% ± 0.8	93.6% ± 0.7	93.4
Microcalcification	93.9% ± 0.8	93.6% ± 0.7	93.4

aim to demonstrate that models with synthetic datasets perform accurately and do not overfit. In Table 8, we show the accuracy metrics across training, validation, and testing for the best-balancing strategies in each dataset. The results show the average values and standard deviations after applying k-fold validation.

For all the datasets, the MLPs obtain good results, as they achieve a trade-off between bias and variance [46]. In terms of bias, the values of the metrics are sufficient. In terms of variance, the differences between training, validation, and testing are low in most cases but never too high. In terms of standard deviation, we can conclude that all the models are very stable. The experimental results presented in Table 8 indicate that the synthetic data are sufficient, as the MLPs obtain valid values.

Although, we acknowledge that SOMs incur a higher computational cost during the training phase; however, once trained, they classify data efficiently with negligible computational overhead. Our approach evaluates imbalanced strategies by measuring classification efficiency after SOM training, thereby maintaining manageable execution times.

This method provides a clear computational advantage compared to traditional techniques, which typically require model retraining or repeated execution for each balancing strategy, resulting in considerably higher computational demands. In contrast, our approach significantly reduces the need for frequent retraining, enhancing computational efficiency. Following, we present Tables 9 and 10 which present some information about the computational cost of our method.

As we can see training all the maps after applying all the grid search combination are not very costly, but in terms of classifying new data for evaluating different strategies for creating synthetic data are very low.

Table 9 Grid search Times for each dataset

Dataset	Grid search time	Train time	Average train time	Dataset Size
Breast cancer	17′ 58″	13′ 43″	1.45′ ± 0.45	306 × 2
Oil spill	30′ 28″	18′ 47″	1.98′ ± 0.66	937 × 48
German credit	28′ 04″	16′ 30″	1.72′ ± 0.55	1000 × 19
Phoneme	1 h 15′ 28″	13′ 51″	1.46′ ± 0.45	5404 × 4
Microcalcification	2 h 39′ 26″	12′ 06″	1.21′ ± 0.45	11,183 × 6
Credit fraud	2 h 34′ 55″	10′ 44″	0.81′ ± 0.52	10,807 × 29

Table 10 Classification time for each minority strategy combination in each dataset

Strategy	Oil Spill	Breast Cancer	German Credits	Phonemes	Microcalcifications	Credit card Fraud
SMOTE	7.42″	2.46″	5.58″	39.80″	2′ 9.93″	1′ 59.10″
ADASYN	8.06″	2.59″	5.60″	39.29″	2′ 13.32″	1′ 48.06″
BSMOTE	7.35″	2.22″	5.77″	39.44″	2′ 15.30″	1′ 56.70″
SVM SMOTE	8.42″	2.32″	7.10″	41.43″	2′ 14.81″	2′ 4.12″
KSMOTE	9.20″	2.59″	9.10″	45.66″	2′ 28.93″	2′ 0.25″
Total:	40.45″	12.18″	33.15″	3′ 25.62″	11′ 22.29″	9′ 48.23″

Conclusions and future work

This paper introduces a methodological approach using Kohonen maps to evaluate various imbalanced data strategies. We applied a combination of five oversampling and undersampling techniques to generate synthetic data, resulting in a total of twenty-five different methods. Initially, we assessed the performance of these strategies via two SOM metrics: topological and quantization errors. These metrics, derived from training and applying the strategies to six different datasets, indicated which strategies performed better. Given the minimal differences between these errors, we introduced a new metric based on the topological properties of Kohonen maps, which was applied to the best results obtained thus far. This metric was applied to all the strategies across the six datasets, and its potential was demonstrated by training six MLPs (one for each dataset) using the best-performing imbalanced data strategies according to our metric. The main strength of this metric is that it considers the original data, so the best strategy is the one that generates synthetic data that more reflects the nature of the problem. In state-of-the-art approaches, this selection is based on the performance of the ML models, which could achieve better performance with instances that are not similar to those generated for the use case.

The main limitation of this study is the variation in the number of imbalanced instances between classes within the datasets. Additionally, the datasets differ in total instances and the number of features per individual. The main limitation of this study is the exclusive use of six binary, numeric, and complete datasets, which may restrict the generalizability of the results. The absence of multiclass datasets, as well as hybrid and/or incomplete datasets, limits the ability to assess the effectiveness of the proposed SOM-based metric across a broader range of data scenarios.

In future work, we aim to apply this workflow to real-world cases where data imbalance arises due to scarcity. By generating synthetic data to balance these datasets, we

hope to improve the performance of classifiers that previously struggled with imbalanced data. We also want to replicate the workflow but using newer undersampling and oversampling techniques proposed in recent years.

Appendix A: Sensitivity analysis of the grid search in the six studied datasets

See Fig. 4, 5, 6, 7, 8, 9

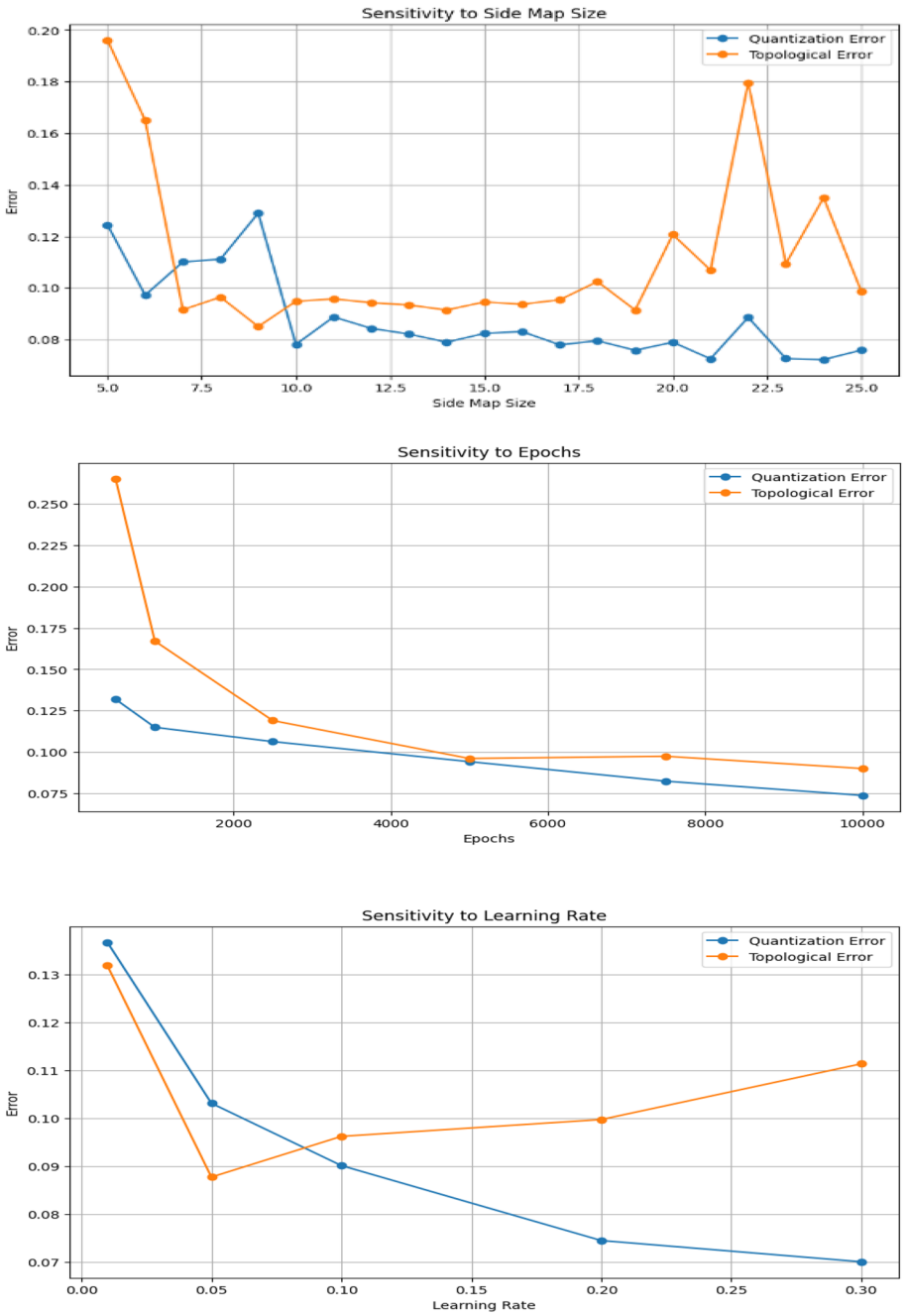

Fig. 4 Quantization error (blue) and Topological error (orange) throughout the grid search for the side map size (left), epochs (middle) and learning rate (right) for the breast cancer dataset

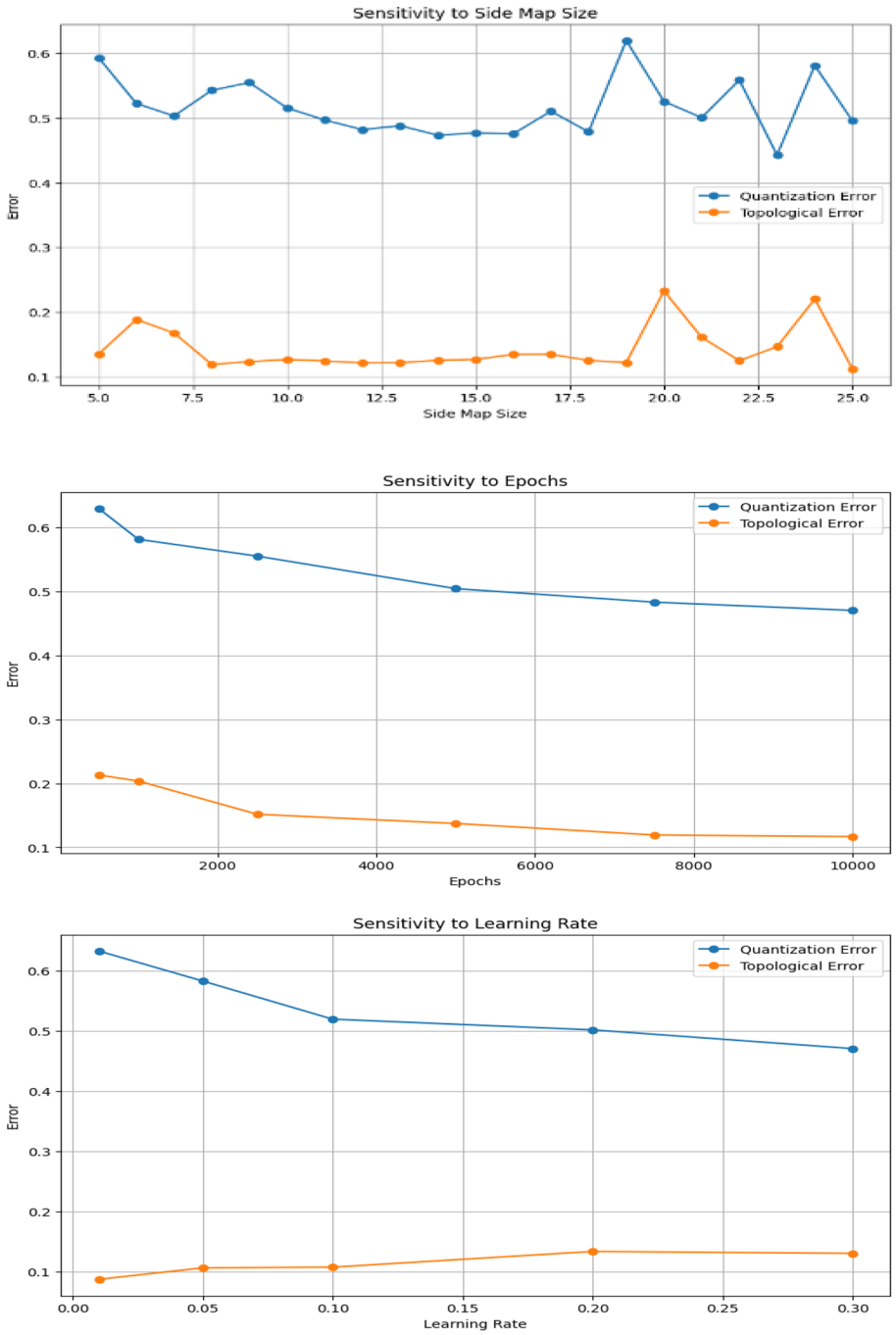

Fig. 5 Quantization error (blue) and Topological error (orange) throughout the grid search for the side map size (left), epochs (middle) and learning rate (right) for the oil Spill dataset

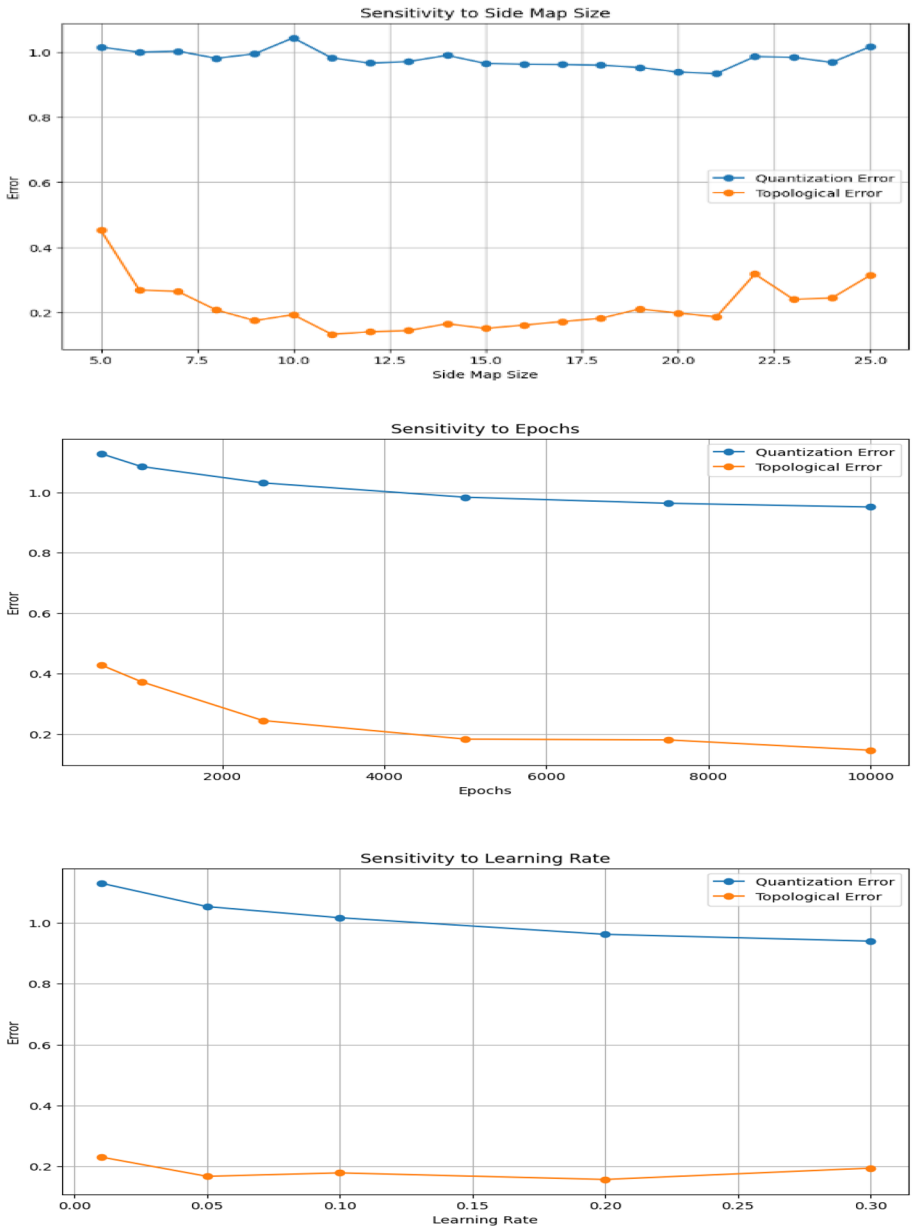

Fig. 6 Quantization error (blue) and Topological error (orange) throughout the grid search for the side map size (left), epochs (middle) and learning rate (right) for the german credit dataset

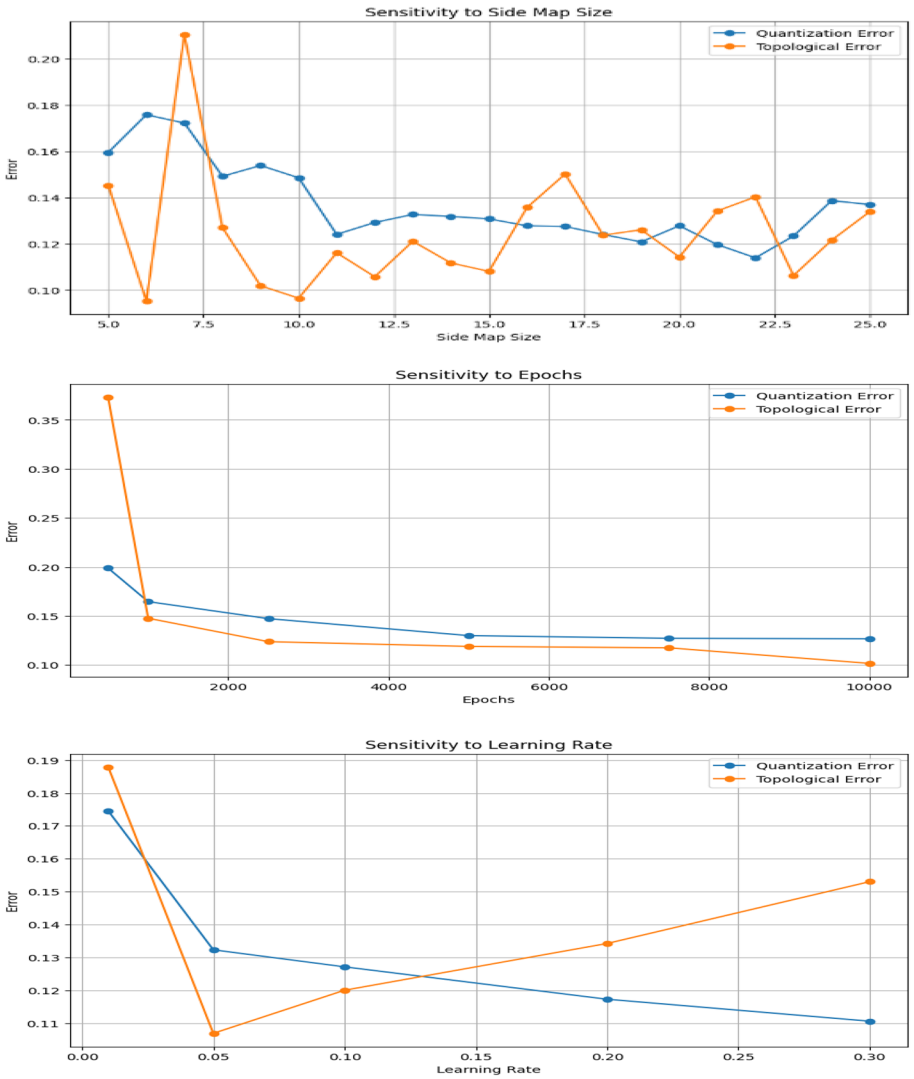

Fig. 7 Quantization error (blue) and Topological error (orange) throughout the grid search for the side map size (left), epochs (middle) and learning rate (right) for the phoneme dataset

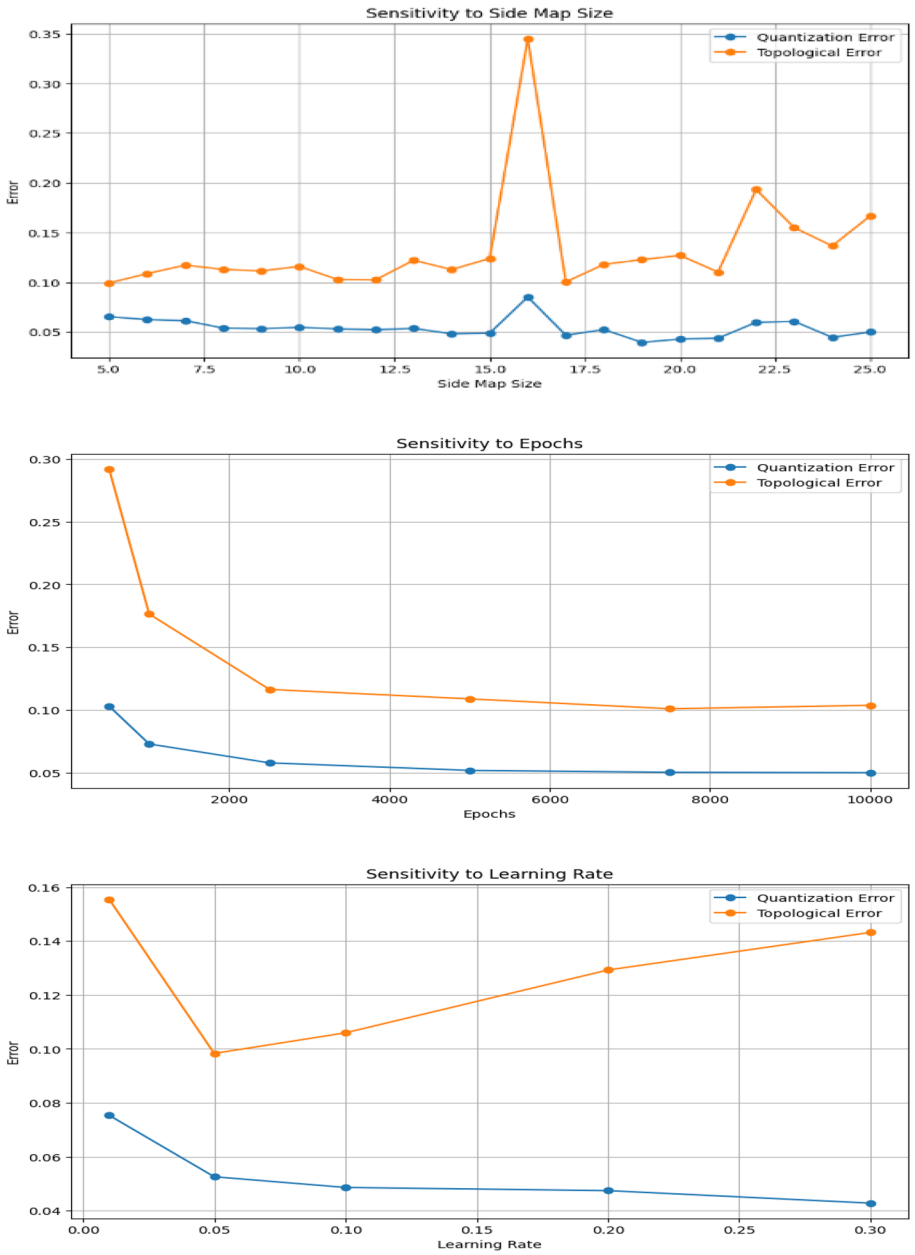

Fig. 8 Quantization error (blue) and Topological error (orange) throughout the grid search for the side map size (left), epochs (middle) and learning rate (right) for the microcalcification dataset

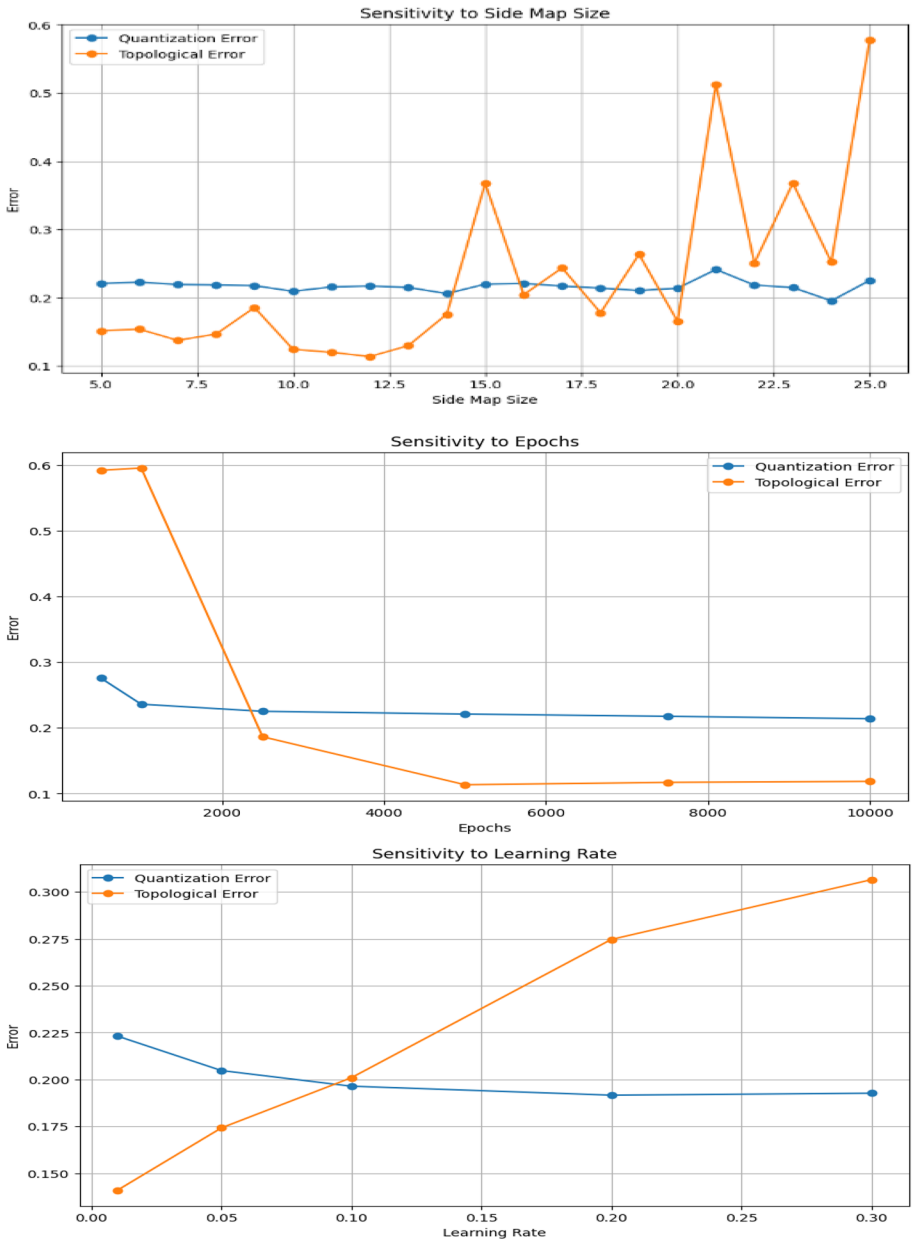

Fig. 9 Quantization error (blue) and Topological error (orange) throughout the grid search for the side map size (left), epochs (middle) and learning rate (right) for the credit card Fraud dataset

Appendix B: Raw results for all experiments from the six studied datasets

See Tables 11, 12, 13, 14, 15, 16

Table 11 Kohonen Map error for bank loans dataset

		Quantization error	Topographic error
SMOTE	Tomek Links	0.926	0.176
	Edited Nearest Neighbours	**0.916**	**0.174**
	Condensed Nearest Neighbours	**0.912**	**0.172**
	Neighbourhood Cleaning Rule	**0.914**	**0.179**
	One Side Selection	0.920	0.190
ADASYN	Tomek Links	0.923	0.177
	Edited Nearest Neighbours	0.924	0.183
	Condensed Nearest Neighbours	**0.912**	**0.172**
	Neighbourhood Cleaning Rule	**0.914**	**0.179**
	One Side Selection	0.920	0.190
Borderline SMOTE	Tomek Links	0.926	0.175
	Edited Nearest Neighbours	0.915	0.167
	Condensed Nearest Neighbours	0.919	0.163
	Neighbourhood Cleaning Rule	0.923	0.175
	One Side Selection	0.924	0.170
SVM SMOTE	Tomek Links	0.954	0.175
	Edited Nearest Neighbours	0.974	0.200
	Condensed Nearest Neighbours	0.968	0.170
	Neighbourhood Cleaning Rule	0.960	0.184
	One Side Selection	0.978	0.170
K-Means SMOTE	Tomek Links	0.940	0.163
	Edited Nearest Neighbours	0.933	0.174
	Condensed Nearest Neighbours	0.936	0.163
	Neighbourhood Cleaning Rule	0.964	0.171
	One Side Selection	0.939	0.162

Bold values represent the best-performing strategy for each dataset

Table 12 Kohonen Map error for phonemes dataset

		Quantization error	Topographic error
SMOTE	Tomek Links	0.143	0.226
	Edited Nearest Neighbours	0.142	0.215
	Condensed Nearest Neighbours	0.143	0.216
	Neighbourhood Cleaning Rule	0.143	0.217
	One Side Selection	0.145	0.221
ADASYN	Tomek Links	**0.142**	**0.212**
	Edited Nearest Neighbours	**0.141**	**0.207**
	Condensed Nearest Neighbours	0.143	0.216
	Neighbourhood Cleaning Rule	0.143	0.217
	One Side Selection	0.145	0.221
Borderline SMOTE	Tomek Links	**0.145**	**0.206**
	Edited Nearest Neighbours	0.143	0.213
	Condensed Nearest Neighbours	**0.144**	**0.206**
	Neighbourhood Cleaning Rule	**0.141**	**0.208**
	One Side Selection	0.146	0.219
SVM SMOTE	Tomek Links	0.145	0.217
	Edited Nearest Neighbours	0.143	0.211
	Condensed Nearest Neighbours	0.145	0.215
	Neighbourhood Cleaning Rule	0.145	0.220
	One Side Selection	0.146	0.215
K-Means SMOTE	Tomek Links	0.165	0.240
	Edited Nearest Neighbours	0.150	0.248
	Condensed Nearest Neighbours	0.151	0.246
	Neighbourhood Cleaning Rule	0.157	0.239
	One Side Selection	0.218	0.152

Bold values represent the best-performing strategy for each dataset

Table 13 Kohonen Map error for cancer breast dataset

		Quantization error	Topographic error
SMOTE	Tomek Links	0.081	0.118
	Edited Nearest Neighbours	**0.076**	**0.113**
	Condensed Nearest Neighbours	0.080	0.163
	Neighbourhood Cleaning Rule	0.080	0.128
	One Side Selection	**0.076**	**0.105**
ADASYN	Tomek Links	0.081	0.114
	Edited Nearest Neighbours	0.077	0.118
	Condensed Nearest Neighbours	**0.080**	**0.086**
	Neighbourhood Cleaning Rule	0.080	0.128
	One Side Selection	**0.076**	**0.105**
Borderline SMOTE	Tomek Links	0.079	0.126
	Edited Nearest Neighbours	0.081	0.117
	Condensed Nearest Neighbours	0.078	0.120
	Neighbourhood Cleaning Rule	0.077	0.134
	One Side Selection	0.085	0.121
SVM SMOTE	Tomek Links	0.100	0.121
	Edited Nearest Neighbours	0.097	0.128
	Condensed Nearest Neighbours	0.083	0.109
	Neighbourhood Cleaning Rule	0.092	0.116
	One Side Selection	0.097	0.102
K-Means SMOTE	Tomek Links	0.084	0.118
	Edited Nearest Neighbours	0.140	0.125
	Condensed Nearest Neighbours	0.094	0.106
	Neighbourhood Cleaning Rule	0.095	0.138
	One Side Selection	**0.079**	**0.103**

Bold values represent the best-performing strategy for each dataset

Table 14 Kohonen Map error for credit frauds dataset

		Quantization error	Topographic error
SMOTE	Tomek Links	0.406	0.074
	Edited Nearest Neighbours	0.407	0.074
	Condensed Nearest Neighbours	0.408	0.075
	Neighbourhood Cleaning Rule	0.407	0.073
	One Side Selection	0.405	0.075
ADASYN	Tomek Links	**0.221**	**0.156**
	Edited Nearest Neighbours	**0.221**	**0.158**
	Condensed Nearest Neighbours	0.225	0.157
	Neighbourhood Cleaning Rule	**0.221**	**0.157**
	One Side Selection	0.223	0.158
Borderline SMOTE	Tomek Links	0.243	0.144
	Edited Nearest Neighbours	0.242	0.149
	Condensed Nearest Neighbours	0.243	0.145
	Neighbourhood Cleaning Rule	0.242	0.148
	One Side Selection	0.243	0.149
SVM SMOTE	Tomek Links	0.282	0.149
	Edited Nearest Neighbours	0.292	0.149
	Condensed Nearest Neighbours	0.277	0.148
	Neighbourhood Cleaning Rule	0.269	0.149
	One Side Selection	0.289	0.156
K-Means SMOTE	Tomek Links	0.313	0.230
	Edited Nearest Neighbours	0.295	0.133
	Condensed Nearest Neighbours	0.231	0.174
	Neighbourhood Cleaning Rule	**0.239**	**0.112**
	One Side Selection	**0.196**	**0.152**

Bold values represent the best-performing strategy for each dataset

Table 15 Kohonen Map error for oil spills dataset

		Quantization error	Topographic error
SMOTE	Tomek Links	0.563	0.112
	Edited Nearest Neighbours	0.566	0.114
	Condensed Nearest Neighbours	0.566	0.119
	Neighbourhood Cleaning Rule	0.556	0.116
	One Side Selection	0.555	0.120
ADASYN	Tomek Links	0.529	0.184
	Edited Nearest Neighbours	0.528	0.192
	Condensed Nearest Neighbours	0.541	0.161
	Neighbourhood Cleaning Rule	**0.530**	**0.175**
	One Side Selection	0.538	0.157
Borderline SMOTE	Tomek Links	0.554	0.179
	Edited Nearest Neighbours	0.555	0.178
	Condensed Nearest Neighbours	0.561	0.171
	Neighbourhood Cleaning Rule	0.556	0.177
	One Side Selection	0.558	0.174
SVM SMOTE	Tomek Links	0.579	0.145
	Edited Nearest Neighbours	0.601	0.102
	Condensed Nearest Neighbours	0.592	0.09
	Neighbourhood Cleaning Rule	0.597	0.105
	One Side Selection	0.593	0.114
K-Means SMOTE	Tomek Links	0.537	0.154
	Edited Nearest Neighbours	**0.513**	**0.146**
	Condensed Nearest Neighbours	**0.538**	**0.138**
	Neighbourhood Cleaning Rule	**0.525**	**0.133**
	One Side Selection	**0.538**	**0.145**

Bold values represent the best-performing strategy for each dataset

Table 16 Kohonen Map error for microcalcifications dataset

		Quantization error	Topographic error
SMOTE	Tomek Links	0.095	0.208
	Edited Nearest Neighbours	0.096	0.213
	Condensed Nearest Neighbours	0.092	0.202
	Neighbourhood Cleaning Rule	0.096	0.218
	One Side Selection	0.093	0.205
ADASYN	Tomek Links	0.073	0.194
	Edited Nearest Neighbours	0.075	0.203
	Condensed Nearest Neighbours	0.076	0.202
	Neighbourhood Cleaning Rule	0.075	0.199
	One Side Selection	0.073	0.196
Borderline SMOTE	Tomek Links	0.078	0.252
	Edited Nearest Neighbours	0.077	0.244
	Condensed Nearest Neighbours	0.077	0.248
	Neighbourhood Cleaning Rule	0.077	0.245
	One Side Selection	0.078	0.243
SVM SMOTE	Tomek Links	**0.088**	**0.107**
	Edited Nearest Neighbours	**0.092**	**0.101**
	Condensed Nearest Neighbours	**0.093**	**0.098**
	Neighbourhood Cleaning Rule	**0.088**	**0.108**
	One Side Selection	**0.091**	**0.109**
K-Means SMOTE	Tomek Links	0.193	0.056
	Edited Nearest Neighbours	0.154	0.177
	Condensed Nearest Neighbours	0.194	0.057
	Neighbourhood Cleaning Rule	0.187	0.108
	One Side Selection	0.145	0.172

Bold values represent the best-performing strategy for each dataset

Abbreviations

ML	Machine learning
SOMs	Self-organizing maps
SMOTE	Synthetic minority over-sampling technique
NCR	Neighbour-cleaning rule
KNN	K-nearest neighbour
SMO	Sequential minimal optimization
NB	Naïve Bayes
ASUWO	Adaptive semisupervised weighted oversampling
DT	Decision tree
RF	Random forest
TL	Tomek links
OSS	One-sided selection
SVM	Support vector machine
RBF	Radial basis function
ENN	Edited nearest neighbour
RUS	Random undersampling strategy
ROS	Random oversampling strategy
CNN	Condensed nearest neighbour
BCBSMOTE	BIRCH clustering, and borderline SMOTE
Gm-SOINN	Gaussian membership-based self-organizing incremental neural network
TLFRNN	Topology learning-based fuzzy random neural network
MLPs	Multilayer perceptrons
PCA	Principal component analysis
ADASYN	Adaptive synthetic sampling
CS	Condensed set

TS	Training set
CNNR	Condensed nearest neighbour rule
OSS	One-sided selection
BMU	Best matching unit
SOM-AGT	Similarity over maps activation-based grid topology

Author contributions
A.N. and A.G.T.: Conceptualization, formal analysis, experimental design and methodology. D.G.: Data curation, creation of SW A.N.: Manuscript writing, figures and tables. All authors reviewed the manuscript. A.G.T.: Approval of the final version.

Funding
Not applicable.

Availability of data and materials
Datasets are freely available. Code has been uploaded to this repository: https://github.com/ufvceiec/SOM_imbalanced

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 1 December 2024 Accepted: 12 May 2025
Published online: 03 June 2025

References

1. Pérez J, Iturbide E, Olivares V, Hidalgo M, Almanza N, Martínez A. A data preparation methodology in data mining applied to mortality population databases. In: Rocha A, Correia AM, Costanzo S, Reis LP, editors. New contributions in information systems and technologies. Cham: Springer International Publishing; 2015. p. 1173–82.
2. Dong Q, Gong S, Zhu X. Imbalanced deep learning by minority class incremental rectification. IEEE Trans Pattern Anal Mach Intell. 2019;41:1367–81.
3. Johnson JM, Khoshgoftaar TM. Survey on deep learning with class imbalance. J Big Data. 2019;6:1–54.
4. Kohonen T. Self-organized formation of topologically correct feature maps. Biol Cybern. 1982;43:59–69.
5. Kohonen T. The self-organizing map. Neurocomputing. 1998;21:1–6.
6. Chawla NV, Lazarevic A, Hall LO, Bowyer KW. SMOTEBoost: improving prediction of the minority class in boosting. In: Lavrač N, Gamberger D, Todorovski L, Blockeel H, editors. Knowledge discovery in databases: PKDD 2003. Berlin, Heidelberg: Springer; 2003. p. 107–19.
7. Junsomboon N, Phienthrakul T. Combining over-sampling and under-sampling techniques for imbalance dataset. In: Proceedings of the 9th international conference on machine learning and computing. Singapore: Association for Computing Machinery. 2017. p. 243–7.
8. Choirunnisa S, Lianto J. Hybrid method of undersampling and oversampling for handling imbalanced data. In, international seminar on research of information technology and intelligent systems (ISRITI). Yogyakart, Indonesia IEEE. 2018;2018:276–80.
9. Wainer J, Franceschinell RA. An empirical evaluation of imbalanced data strategies from a practitioner's point of view. arXiv preprint arXiv:181007168. 2018.
10. Costa AJ, Santos MS, Soares C, Abreu PH. Analysis of imbalance strategies recommendation using a meta-learning approach. In: 7th ICML workshop on automated machine learning (AutoML-ICML2020): ICML; 2020. p. 1–10.
11. Sun A, Lim E-P, Liu Y. On strategies for imbalanced text classification using SVM: a comparative study. Decis Support Syst. 2009;48:191–201.
12. Goel G, Maguire L, Li Y, McLoone S. Evaluation of sampling methods for learning from imbalanced data. In: Huang DS, Bevilacqua V, Figueroa JC, Premaratne P, editors. Intelligent computing theories. Berlin: Springer; 2013.
13. Shamsudin H, Yusof UK, Jayalakshmi A, Khalid MNA. Combining oversampling and undersampling techniques for imbalanced classification: a comparative study using credit card fraudulent transaction dataset. In: 2020 IEEE 16th international conference on control & automation (ICCA). Singapore: IEEE; 2020. p. 803–8.
14. A Gosain S Sardana. 2017. Handling class imbalance problem using oversampling techniques: a review. In, international conference on advances in computing, communications and informatics (ICACCI) Udupi, India: IEEE 2017 79 85
15. Kraiem MS, Sánchez-Hernández F, Moreno-García MN. Selecting the suitable resampling strategy for imbalanced data classification regarding dataset properties an approach based on association models. Appl Sci. 2021;11:8546.
16. Wongvorachan T, He S, Bulut O. A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. Information. 2023;14:54.
17. Mujahid M, Kına E, Rustam F, Villar MG, Alvarado ES, Diez IDLT, et al. Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering. J Big Data. 2024;11:87.

18. Alamri M, Ykhlef M. Hybrid undersampling and oversampling for handling imbalanced credit card data. IEEE Access. 2024;12:14050–60.
19. Parrales-Bravo F, Caicedo-Quiroz R, Tolozano-Benitez E, Gómez-Rodríguez V, Cevallos-Torres L, Charco-Aguirre J, et al. OUCH: oversampling and undersampling cannot help improve accuracy in our bayesian classifiers that predict preeclampsia. Mathematics. 2024;12:3351.
20. Santoso B, Wijayanto H, Notodiputro KA, Sartono B. Synthetic over sampling methods for handling class imbalanced problems: a review. IOP Conf Ser: Earth Environ Sci. 2017;58: 012031.
21. Yang C, Fridgeirsson EA, Kors JA, Reps JM, Rijnbeek PR. Impact of random oversampling and random undersampling on the performance of prediction models developed using observational health data. J Big Data. 2024;11:7.
22. Yu H, Lu J, Zhang G. Online topology learning by a Gaussian membership-based self-organizing incremental neural network. IEEE Trans Neural Netw Learn Syst. 2020;31:3947–61.
23. Yu H, Lu J, Zhang G. Topology learning-based fuzzy random neural networks for streaming data regression. IEEE Trans Fuzzy Syst. 2022;30:412–25.
24. Winston JJ, Turker GF, Kose U, Hemanth DJ. Novel optimization based hybrid self-organizing map classifiers for iris image recognition. Int J Comput Intell Syst. 2020;13:1048–58.
25. Kubat M, Holte RC, Matwin S. Machine learning for the detection of oil spills in satellite radar images. Mach Learn. 1998;30:195–215.
26. Tong L, Yongquan L, Weijian NA. A hybrid strategy for imbalanced classification. In, 3rd symposium on web society. Port Elizabeth: IEEE. 2011;2011:105–10.
27. Chen W, Yang K, Yu Z, Shi Y, Chen CLP. A survey on imbalanced learning: latest research, applications and future directions. Artif Intell Rev. 2024;57:137.
28. Elzain HE, Chung SY, Venkatramanan S, Selvam S, Ahemd HA, Seo YK, et al. Novel machine learning algorithms to predict the groundwater vulnerability index to nitrate pollution at two levels of modeling. Chemosphere. 2023;314: 137671.
29. Wang H, Liu X. Undersampling bankruptcy prediction: Taiwan bankruptcy data. PLoS ONE. 2021;16: e0254030.
30. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57.
31. He H, Bai Y, Garcia EA, Li S. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). Hong Kong: IEEE. 2008;2008:1322–8.
32. Han H, Wang WY, Mao BH. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International Conference on Intelligent Computing; 2005. p. 878–87.
33. Nguyen HM, Cooper EW, Kamei K. Borderline over-sampling for imbalanced data classification. In: Proceedings: fifth international workshop on computational intelligence & applications; 2009. p. 24–9.
34. Last F, Douzas G, Bacao F. Oversampling for imbalanced learning based on K-means and smote. arXiv preprint arXiv:171100837. 2017.
35. Tomek I. An experiment with the edited nearest-neighbor rule. IEEE Trans Syst Man Cybern. 1976. https://doi.org/10.1109/TSMC.1976.4309523.
36. Wilson DL. Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern. 1972. https://doi.org/10.1109/TSMC.1972.4309137.
37. Hart P. The condensed nearest neighbor rule (Corresp.). IEEE Trans Inf Theory. 1968;14:515–6.
38. Laurikkala J. Improving identification of difficult small classes by balancing class distribution. In: Quaglini S, Barahona P, Andreassen S, editors. Artificial intelligence in medicine. Berlin: Springer; 2001. p. 63–6.
39. Kubat M, Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the fourteenth international conference on machine learning (ICML 1997). Nashville, Tennessee: Morgan Kaufmann. 1997. 179–86.
40. Jaccard P. The distribution of the flora in the alpine zone. New Phytol. 1912;11:37–50.
41. Sałabun W, Shekhovtsov A. An innovative drastic metric for ranking similarity in decision-making problems. In: Proceedings of the 18th conference on computer science and intelligence systems. Warsaw, Poland: ACSIS; 2023. p. 731–8.
42. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323:533–6.
43. García-Tejedor ÁJ, Nogales A. An open-source python library for self-organizing-maps. Softw Impacts. 2022;12: 100280.
44. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13:281–305.
45. Khalilia M, Popescu M. Topology preservation in fuzzy self-organizing maps. In: Jamshidi M, Kreinovich V, Kacprzyk J, editors. Advance trends in soft computing. Cham: Springer International Publishing; 2014. p. 105–14.
46. Belkin M, Hsu D, Ma S, Mandal S. Reconciling modern machine-learning practice and the classical bias-variance trade-off. Proc Natl Acad Sci U S A. 2019;116:15849–54.

Publisher's Note