Neural Computing and Applications
https://doi.org/10.1007/s00521-024-10286-z

ORIGINAL ARTICLE

Contrastive dissimilarity: optimizing performance on imbalanced and limited data sets

Lucas O. Teixeira[1] · Diego Bertolini[2] · Luiz S. Oliveira[3] · George D. C. Cavalcanti[4] · Yandre M. G. Costa[1]

Received: 14 March 2024 / Accepted: 29 July 2024

Abstract
A primary challenge in pattern recognition is imbalanced datasets, resulting in skewed and biased predictions. This problem is exacerbated by limited data availability, increasing the reliance on expensive expert data labeling. The study introduces a novel method called contrastive dissimilarity, which combines dissimilarity-based representation with contrastive learning to improve classification performance in imbalance and data scarcity scenarios. Based on pairwise sample differences, dissimilarity representation excels in situations with numerous overlapping classes and limited samples per class. Unlike traditional methods that use fixed distance functions like Euclidean or cosine, our proposal employs metric learning with contrastive loss to estimate a custom dissimilarity function. We conducted extensive evaluations in 13 databases across multiple training–test splits. The results showed that this approach outperforms traditional models like SVM, random forest, and Naive Bayes, particularly in settings with limited training data.

Keywords Imbalanced learning · Dissimilarity · Contrastive learning · Metric learning

1 Introduction

A primary challenge in pattern recognition is the effective management of imbalanced datasets [21] characterized by uneven class representation, frequently resulting in skewed and biased model predictions. This challenge intensifies in scenarios with limited data availability, adding complexity to the training process and increasing the dependency on expert data labeling, which is time-consuming and costly [8, 47]. In crucial applications such as medical imaging or autonomous driving, the cost–accuracy trade-off becomes extremely delicate.

Particularly in the context of tabular data, class imbalance remains an important challenge [17]. To mitigate this, various sampling strategies, such as downsampling and oversampling, are frequently employed. Downsampling reduces the majority class data, which may not be a good option if data are already scarce. Oversampling techniques augment the size of minority classes by generating new data points randomly or using more sophisticated options such as Synthetic Minority Over-sampling Technique (SMOTE) [4].

Dissimilarity-based representation offers an alternative feature space by emphasizing the differences between

✉ Lucas O. Teixeira
pg54804@uem.br; lucasxteixeira@gmail.com

Diego Bertolini
diegobertolini@utfpr.edu.br

Luiz S. Oliveira
luiz.oliveira@ufpr.br

George D. C. Cavalcanti
gdcc@cin.ufpe.br

Yandre M. G. Costa
yandre@din.uem.br

[1] Departamento de Informática, Universidade Estadual de Maringá (UEM), Maringá, Paraná, Brazil

[2] Departamento Acadêmico de Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR), Campo Mourão, Paraná, Brazil

[3] Departamento de Informática, Universidade Federal do Paraná (UFPR), Curitiba, Paraná, Brazil

[4] Centro de Informática, Universidade Federal de Pernambuco (UFPE), Recife, Pernambuco, Brazil

Published online: 12 August 2024

individual data samples [37]. This approach has demonstrated efficacy under several conditions: constrained sample sizes [11], high-dimensional feature spaces [41], overlapping class distributions [33, 36, 40], and imbalanced data sets [51].

Typically, two primary techniques for dissimilarity classification are dissimilarity space and the dissimilarity vector [9]. The former is valued for its simplicity and is suitable where class organization is inherently clear, but feature characterization is challenging. The latter transforms multi-class problems into binary ones, proving useful in scenarios with multiple classes or limited instances per class.

To address the shortcomings of traditional dissimilarity space, which depend on predefined metrics like Euclidean distance or cosine similarity, and the simplistic nature of dissimilarity vector that rely on difference vectors, we propose a new approach: a task-specific metric learning strategy via a contrastive learning framework to address this limitation.

Contrastive learning has garnered significant interest for its efficacy in self-supervised learning tasks [10, 15]. The central idea revolves around learning embeddings that bring similar data points closer together in the feature space while pushing dissimilar points farther apart. This approach identifies unique features of each data sample, enhancing its ability to differentiate between various classes or clusters. In supervised learning, it is beneficial when there is a shortage of labeled data, leading to improved generalization [19]. The method has gained popularity recently, notably due to the introduction of the SimCLR framework [5].

This research presents a novel methodology that synergizes dissimilarity-based representation with contrastive learning. The hypothesis is that focusing on the differences between data samples using a dissimilarity approach, coupled with the contrastive loss that highlights both similarities and differences, can help us better classify data, especially when dealing with scarce and imbalanced datasets. To validate this hypothesis, we conduct extensive evaluations of the model across multiple training–test splits to examine the impact of constrained training sets on the model performance.

We conducted extensive comparisons across 13 databases, employing 10 different training–test splits, varying in the amount of training data. Our findings demonstrate the superiority of our proposed approach over traditional models such as SVM, random forest, and Naive Bayes; our contrastive dissimilarity space model achieved a 60% win rate with approximately 25% draws, while the contrastive dissimilarity vector model achieved around 40% wins and 15% draws. The performance difference increases with more limited splits, with a combined win and draw rate of 94.9% and 69.2%, respectively.

The remaining sections of this work are as follows: Sect. 1 surveys literature pertinent to our study, Sect. 2 describes the methodology we adopt, termed as contrastive dissimilarity, Sect. 3 elaborates on the experimental framework, encompassing the database, algorithms, and parameters used, Sect. 4 provides an analysis of the findings, and Sect. 5 offers concluding remarks and potential avenues for future research.

2 Literature review

In this section, we examine key concepts central to our paper and highlight their recent progress, including dissimilarity-based classification, metric learning, and contrastive learning.

2.1 Dissimilarity

Dissimilarity-based classification is effective for complex tasks with limited training data [37]. It comes in two forms: dissimilarity space and vectors. The former uses a matrix to compare training samples with prototypes [37], while the latter employs standard classifiers to evaluate the difference between two samples based on their class [3]. The approach has wide applications, including classifying handwritten text [38], content-based image retrieval (CBIR) [31], human-pose estimation [45], and text categorization [39], among others.

Deep learning has also influenced the field, introducing meta-strategies like siamese networks and fixed distance functions for better classification, including classification of spectrograms [27, 28, 52], brain images [1], handwriting [42], and person re-identification [25, 46].

2.2 Metric learning

Metric learning focuses on creating a custom distance metric between samples [22]. While traditional methods like Euclidean and cosine distance metrics are widely used, they may not be suitable for problems involving nonlinear relationships or high-dimensional data. Custom metrics can be generated using supervised or semi-supervised approaches.

In the context of dissimilarity, [29] used a siamese network and metric learning for image classification, achieving state-of-the-art results. [30] used the triplet loss function and explored techniques for generating dissimilarity space, showing improved performance over previous methods.

2.3 Contrastive learning

Contrastive learning is a self-supervised learning approach that trains models to differentiate between similar and dissimilar data points. Originally developed in the context of information theory [15], it has gained significant traction in computer vision, natural language processing, and other domains. The objective is to learn representations such that similar samples are pulled closer in the embedding space while dissimilar ones are pushed apart [32].

A particularly noteworthy aspect of contrastive learning is its effectiveness in dealing with imbalanced data sets. It addresses this challenge by improving the representation of minority classes in the learned embedding space without adversely affecting the representation of the majority classes [23, 24].

The approach has also been adapted for tabular data, particularly in medical and financial applications [6, 49]. These adaptations often involve specialized architectures and data preprocessing methods tailored to the unique characteristics of tabular data. Despite the differences from image or text data, contrastive learning in this context has shown promising results in tasks like anomaly detection and clustering [14, 48].

3 Proposed method

The proposed method is composed of three phases: metric learning (Phase 1), siamese network training (Phase 2), and dissimilarity representation (Phase 3). In Phase 1, the focus is on the architectural specifics prior to training, primarily concerning the projection head, which serves as the metric learning component.

Phase 2 involves the siamese network training; considering tabular data, a row is paired with another from the same class. This process is repeated to create the number of pairs specified by the batch size. For each pair, the absolute difference between them is then fed into the projection head, which generates a dissimilarity value used for calculating the contrastive loss. Internally, the procedure automatically constructs all possible pairings among the provided samples, thereby eliminating the need for explicitly labeled negative pairs.

Phase 3 is dedicated to dissimilarity representation; the projection head is utilized to estimate dissimilarity values, replacing the need for a predetermined distance function. It involves calculating the dissimilarity of each data observation against a set of prototypes to form both the training and testing sets. The prototypes are a selected subset of representative samples.

3.1 Phase 1: metric learning

This study introduces a novel metric learning approach that leverages contrastive learning to create a task-specific measure of dissimilarity for tabular data. Our method goes beyond traditional distance measures like Euclidean or cosine distances [35]. It learns to identify differences between data pairs during training, as illustrated in Fig. 1.

The projection head consists of several fully connected layers that take the absolute difference between two data points as input and compute their dissimilarity. The dissimilarity value is then used to determine the contrastive loss. During the learning process, the projection head is trained to increase the dissimilarity for data point pairs belonging to different classes and decrease it for pairs from the same class.

The architecture of the projection head plays a crucial role in shaping the mapping function. A single-layer projection head generates a linear mapping, whereas a multilayer structure enables a nonlinear function.

3.2 Phase 2: siamese network training

Algorithm 1 outlines the procedure for training our contrastive dissimilarity model. The model uses the NT-Xent loss function as its loss [5], eliminating the need for labeling negative pairs. Unlike the original model, which only handles one positive pair, we modified the loss function to accommodate multiple positive pairs. The input comprises two sets, d' and d'', each containing data points. The batch size specifies the number of pairs within each set. Each pair is matched so that they belong to the same class. This means the first element in d'_1 corresponds to the same class as the first element in d''_1, and this pattern continues for subsequent elements in both sets. The first step (Line 1) concatenates d' and d'' into a single set x. Following this, Lines 2–3 employ the tile and repeat operations to expand x into x' and x''. Tile replicates an array along specified axes, creating a larger array by repeating the original pattern, while repeat duplicates each element of an array a specified number of times, resulting in an expanded array where elements are repeated individually. This expansion serves a specific purpose: it generates all feasible pair combinations between d' and d''. In Line 4, the algorithm computes the dissimilarity for all positive and negative pairs. Line 5 then reshapes this set into a matrix format, aligning it with the input shape expected by the NT-Xent loss. Finally, Line 6 calculates the NT-Xent loss. The loss function aims to minimize the dissimilarity for positive pairs while maximizing it for negative pairs.

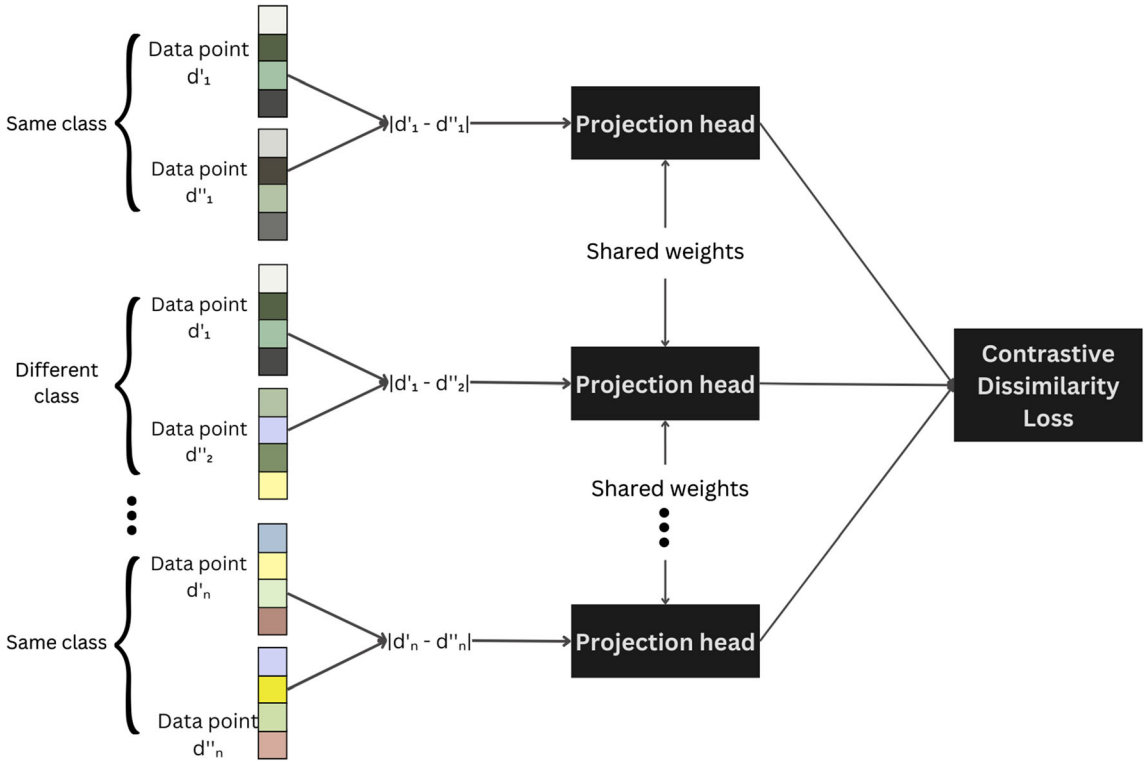


Fig. 1 Contrastive dissimilarity training schema

Algorithm 1 Contrastive dissimilarity training.

Input: input d' and d'', labels y, batch size n, model m
Output: updated model m
1: $x \leftarrow concatenate(d', d'')$
2: $x' \leftarrow tile(x, n * 2)$
3: $x'' \leftarrow repeat(x, n * 2)$
4: $D \leftarrow m.diss(x' - x'')$
5: $D \leftarrow reshape(D, (n * 2, -1))$
6: $L \leftarrow contrastive\ loss(D, y)$
7: $m.backward(L)$
8: **return** m

Algorithm 2 Dissimilarity NT-Xent Loss

Input: dissimilarity matrix $diss$, labels y
Output: contrastive loss
1: $size \leftarrow sizeof(diss)$
2: $y \leftarrow concatenate(y, y)$
3: $y' \leftarrow tile(y, size)$
4: $y'' \leftarrow repeat(y, size)$
5: $pos\_mask \leftarrow reshape(y' == y'', (size, size))$
6: set diagonal of pos_mask to False
7: $neg\_mask \leftarrow$ inverted identity matrix with size $= size$
8: $nominator \leftarrow sum(pos\_mask \cdot exp(diss/t), axis = 1)$
9: $denominator \leftarrow sum(neg\_mask \cdot exp(diss/t), axis = 1)$
10: $loss\_partial \leftarrow -log(nominator/denominator)$
11: $loss \leftarrow mean(loss\_partial)$
12: **return** $loss$

Algorithm 2 outlines the pseudocode for computing the dissimilarity NT-Xent loss. The function takes two inputs: the dissimilarity matrix *diss* and the labels y. Additionally, there is a temperature parameter t set during initialization. First, in lines 2–6, a positive mask is created to account for multiple positive pairs, followed by the creation of a negative mask in line 7. In lines 8 and 9, the numerator and denominator are computed respectively, then divided, and followed by taking the negative logarithm. Finally, the average of the results is calculated to obtain the loss. This approach is based on the original work by [5].

The data are composed of pairs selected in a two-step random process. First, a class is chosen at random, with no consideration given to its frequency in the overall dataset, and then two unique instances are randomly selected. This ensures equal representation for each class, regardless of its prevalence in the total dataset.

3.3 Phase 3: dissimilarity representation

In feature space representation, dissimilarity focuses on highlighting the differences between samples, and it is

advantageous when conventional features fall short in differentiating samples. This is often the case in large-scale multi-class issues or situations where only a few samples are available for each class. Dissimilarity can be categorized into two main types: dissimilarity spaces and vectors.

3.3.1 Dissimilarity space

Let us define T as a training set containing n samples and R as the prototype set with m samples. Ideally, it would be optimal to employ every available sample as a prototype, but this could require extensive or impractical computational resources. The goal of prototype selection is to simplify the issue by choosing a subset of training samples that adequately represent the whole. The $D(T,R)$ represents dissimilarity matrix and is formulated as:

$$D(T,R) = \begin{bmatrix} v(x_1,p_1) & v(x_1,p_2) & \dots & v(x_1,p_m) \\ v(x_2,p_1) & v(x_2,p_2) & \dots & v(x_2,p_m) \\ \vdots & \vdots & \ddots & \vdots \\ v(x_n,p_1) & v(x_n,p_2) & \dots & v(x_n,p_m) \end{bmatrix}$$

where x_i is the ith sample in the training set, p_j is the jth prototype, and v is the dissimilarity function. Each row in this matrix corresponds to a training sample, while each column is associated with a prototype. A conventional classification model can be trained using $D(T,R)$.

For the testing phase, assume t_k is the kth sample in the testing set. The testing vector $D'(t_k,R)$ can be expressed as:

$$D'(t_k,R) = [\, v(t_k,p_1) \quad v(t_k,p_2) \quad \dots \quad v(t_k,p_m) \,]$$

where t_k and p_j are the kth testing sample and jth prototype, respectively. The resulting $D'(t_k,R)$ has a column count matching $D(T,R)$, allowing the usage of the trained classification model to determine class probabilities.

3.3.2 Dissimilarity vector

This proposal uses a metric learning approach that outputs a single-value dissimilarity between two samples rather than a vector. To boost robustness, we adopted a strategy similar to that of [13], where dropout is enabled, resulting in w predictions for each pair, expanding the dimensions of the resulting vectors. Given x_{ij} as the ith class, jth sample, p_{ik} as the ith class, kth prototype, and v as the dissimilarity function, the resulting vector $v'(x_{ij},p_{ik})$ is defined as:

$$v'(x_{ij},p_{ik}) = [\, v(x_{ij},p_{ik})_1 \quad \dots \quad v(x_{ij},p_{ik})_w \,]$$

The training set T includes both positive $T_\oplus$ and negative $T_\ominus$ pairs. The vector is marked as positive if the sample and prototype belong to the same class; otherwise, it is negative. With n classes, m samples, and m' prototypes per class, these sets are defined as:

$$T_\oplus = v'(x_{ij},p_{ik}) \text{ where}$$
$$i = 1 \text{ to } n,\ j = 1 \text{ to } m,\ k = 1 \text{ to } m'$$

$$T_\ominus = v'(x_{ij},p_{kl}) \text{ where}$$
$$i,k = 1 \text{ to } n,\ i \neq k,\ j = 1 \text{ to } m,\ l = 1 \text{ to } m'$$

During testing, dropout is enabled to generate w dissimilarity values between each test sample and prototype. This is done for all ik training prototypes, with i representing the number of classes and k the number of prototypes in each class, resulting in ik separate feature vectors. The trained model then uses these vectors to make the final predictions. Each prediction gives a specific probability, showing how likely the test sample is to be from the same class as the associated prototype.

4 Experimental setup

This section presents details regarding the experimental setup adopted in this work, including datasets, training, and evaluation protocols.

4.1 Databases

We employed 11 databases from Keel repository [2] and 2 databases from UC Irvine Machine Learning Repository [18]. We focused primarily on datasets with an imbalance ratio higher than five, meaning that the majority class is five times more frequent than the minority class, the only exceptions being Hayes-Roth and Glass1. Table 1 presents the datasets with their main characteristics.

4.2 Training protocol

To enhance the efficacy of the contrastive learning procedure, a strategic augmentation was employed within each cross-validation fold of the training set. This augmentation involved utilizing the Synthetic Minority Over-sampling Technique (SMOTE) to achieve a balanced class distribution in the training set, addressing class imbalance issues.

The augmentation protocol was strictly applied during the training phase of the contrastive dissimilarity model and the prototype selection process. In these stages, leveraging an enriched dataset was important for the development of a robust model. However, it's important to note that for critical steps like the generation of the dissimilarity matrix and the selection of sample pairs for constructing the dissimilarity vector, the original, unmodified dataset was exclusively used.

Table 1 Databases

No	Source	Database	Instances	Features	Classes	Imbalance ratio
01	Keel	Hayes-Roth	132	4	3	1.70
02	Keel	Glass1	214	9	2	1.82
03	Keel	New-Thyroid	215	5	3	5.00
04	Keel	Dermatology	366	34	6	5.60
05	Keel	Balance	625	4	3	5.88
06	Keel	Segment0	2308	19	2	6.02
07	UCI	DryBean [20]	13,611	16	7	6.79
08	Keel	Glass	214	9	6	8.44
09	Keel	Page-Blocks0	5472	10	2	8.79
10	Keel	Vowel0	988	13	2	9.98
11	Keel	Yeast4	1484	8	2	28.10
12	UCI	HCV [16]	615	12	5	76.14
13	Keel	Yeast	1484	8	10	92.60

We employed a two-step training protocol in our experiments: initialization and fine-tuning, as outlined in Table 2. We set the batch sizes at 64 for the initialization phase and 32 for the fine-tuning phase. In early folds, using larger batch sizes was not feasible due to the small amount of training data.

During the initialization phase, we conducted 10,000 iterations at a learning rate of 10^{-4} to expedite the warm-up period. For the fine-tuning phase, we first carried out 20,000 iterations with a learning rate of 10^{-5} and a temperature setting of 0.5. Followed by another 20,000 iterations at a reduced temperature of 0.1. The rationale behind this was to gradually refine the network weights, initially allowing broader adjustments at a higher temperature and then focusing on more precise and fine-grained steps.

4.3 Evaluation protocol

Our study follows a four-step evaluation protocol: cross-validation, prototype selection, dissimilarity representation, and classification.

We performed a comprehensive analysis to evaluate the performance under different data availability conditions. First, a consistent test set was established, achieved using a stratified holdout to split the data into 70% for training and 30% for testing. The same test set is used across all scenarios, ensuring equitable comparison of results across different folds. Additionally, within the 70% allocated for training, we systematically changed the volume of data utilized for training, ranging incrementally from 10 to 100%; any data not used for training in these scenarios was excluded from the analysis.

Table 2 Training protocol

Step	Iterations	Temperature	Learning rate
Initialization	10,000	1	10^{-4}
Fine-tuning	20,000	0.5	10^{-5}
	20,000	0.1	10^{-5}

For the prototype selection, we employed the K-means clustering algorithm. The algorithm iteratively assigns each data point to the nearest centroid and recalculates the centroids until convergence is achieved. The resulting centroids form the prototype set. To determine the optimal number of prototypes for each task, we trained the models on the last fold of the dataset, adjusting the count of prototypes from two to ten to identify the configuration yielding the highest performance. The last fold contains 100% of the training set, we partitioned this subset in 70% for training and the remaining 30% for evaluation.

Our analysis includes two dissimilarity approaches. The first approach uses a dissimilarity matrix with one row per training sample and one column per prototype. This matrix is suitable for training standard classifiers like support vector machine (SVM) or random forest (RF), for instance. For evaluation, we use testing samples instead of training ones.

The second approach employs a dissimilarity vector, which combines samples and prototypes and considers both positive and negative combinations. We paired each sample with each prototype. Following the same strategy outlined before, we evaluated different values of w for each dataset. Since we are dealing with a relatively small number of classes, the resulting dataset is somewhat balanced.

The final classification is done using scikit-learn [34] with three main classifiers: SVM, random forest, and Naive Bayes. We report only the best F1-Score. For random forest and Naive Bayes, we kept the default

hyperparameters settings. For SVM, we used an RBF kernel and a grid search to find the optimal values of C (0.001, 0.01, 1, 10, 25, 50, 100, 1000) and γ (1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}).

4.4 Projection head

The projection head is a key component in our contrastive dissimilarity model. It shapes how the model measures differences between data points. The more complex the projection head, represented by its depth and density, the better it can identify complex patterns in the data. However, a more complex projection head also risks overfitting.

To find the optimal network architecture, we experimented with settings ranging from a single-layer to six-layer networks, in which the first layer had neuron counts varying from 16 (2^4) to 1024 (2^{10}). In multi-layer setups, each subsequent layer was designed with half the number of neurons compared to the preceding layer. To identify the optimal configuration, we follow the same strategy as before, training the models using the last fold of the dataset, which constituted 100% of the training data, further dividing it into two parts: 70% for training and 30% for evaluation.

5 Results and discussion

In this analysis, we conduct an extensive assessment contrasting our innovative contrastive dissimilarity approach with established machine learning techniques, particularly focusing on their F1-score metrics. The study involves a comparison with three traditional algorithms: support vector machine (SVM), random forest (RF), and Naive Bayes (NB), considering both scenarios with and without Synthetic Minority Over-sampling Technique (SMOTE), as outlined in Table 7.

Table 3 presents our main findings, encompassing the results from our proposed contrastive dissimilarity space (CS) and vector (CV), traditional models (the best of RF, SVM, or NB, abbreviated as Tradt), as well as classic dissimilarity space and vector (DS and DV, respectively).

Our contrastive dissimilarity space model (CS) shows enhanced efficacy over traditional algorithms employing SMOTE, achieving 78 victories (60%), 30 draws (23.1%), and 22 defeats (16.9%). When pitted against traditional methods without SMOTE, it records 75 victories (57.7%), 34 draws (26.2%), and 21 defeats (16.1%). A two-tailed paired t test confirms these differences as statistically significant ($t(129) = 5.24$, $p < .001$; $t(129) = 5.3$, $p < .001$, respectively).

Analyzing the losses more closely, they were primarily in scenarios containing more training data, whereas our model excelled in data-scarce situations. Against traditional models using SMOTE, our approach was outclassed occasionally—thrice by Naive Bayes, twelve times by random forest, and seven times by SVM. Without SMOTE, it was surpassed twice by Naive Bayes, five times by random forest, and fourteen times by SVM.

In datasets limited to the first three folds (30% of total data), our model demonstrates remarkable efficiency. Against traditional models with SMOTE, it notches 32 wins (82.1%), 5 draws (12.8%), and 2 losses (5.1%). Against models without SMOTE, it achieves 30 wins (76.9%), 7 draws (18%), and only 2 losses (5.1%). Statistical analysis via a two-tailed paired t test highlights significant differences in both scenarios ($t(38) = 4.81$ and 4.3, p values $< .001$), with a combined win and draw rate of 94.9% in these limited data cases, surpassing the all-case rates of 83.1 and 83.9%.

The contrastive dissimilarity vector model (CV) slightly lags behind established models using SMOTE, securing 52 wins (40%), 16 draws (12.3%), and 62 losses (47.7%). Against models not using SMOTE, it records 50 wins (38.5%), 23 draws (17.7%), and 57 losses (43.8%). These differences, analyzed through a two-tailed paired t test, are not statistically significant ($t(129) = -0.57$, $p = .57$; $t(129) = -0.55$, $p = .58$).

Focusing on the initial 30 % of the data set, the vector model shows a slight uptick in performance. Against SMOTE-enabled models, it achieves 21 wins (53.8%), 6 draws (15.4%), and 12 losses (30.8%). Without SMOTE, it records 22 wins (56.4%), 5 draws (12.8%), and 12 losses (30.8%). A two-tailed paired t test indicates these results are not significantly different ($t(38) = 2.04$ and 1.85, p values of 0.049 and 0.072). Detailed insights and performance limitations are further discussed in B.

5.1 Classical dissimilarity comparison

In head-to-head comparisons, the contrastive dissimilarity space model significantly outshines conventional models, recording 96 wins (73.8%), 23 draws (17.7%), and just 11 losses (8.5%) ($t(129) = 7.6$, $p < .001$). In limited data scenarios (first three folds), its performance remains exceptional, with 35 wins (89.7%), 4 draws (10.3%), and no losses ($t(38) = 5.86$, $p < .001$).

The vector-based approach also exhibits superior effectiveness over traditional models, securing 82 wins (63.1%), 23 draws (17.7%), and 25 losses (19.2%) ($t(129) = 6.83$, $p < .001$). Its performance further improves in the first three folds, achieving 28 wins (71.8%), 8 draws (20.5%), and 3 losses (7.7%) ($t(38) = 5.1$, $p < .001$).

Table 3 F1-score across different proportions

Database	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Hayes-Roth	CS	0.74	0.76	0.73	0.76	0.76	0.76	0.72	0.78	0.78	0.74
	CS wt/ SMOTE	0.46	0.6	0.7	0.63	0.64	0.69	0.69	0.71	0.69	0.63
	CV	0.78	0.8	0.8	0.78	0.79	0.75	0.75	0.76	0.81	0.79
	CV wt/ SMOTE	0.48	0.6	0.65	0.63	0.65	0.72	0.74	0.76	0.65	0.77
	Tradt	0.51	0.57	0.57	0.58	0.72	0.74	0.78	0.78	0.82	0.8
	Tradt wt/ SMOTE	0.49	0.59	0.6	0.58	0.7	0.75	0.75	0.78	0.77	0.79
	DS	0.54	0.69	0.72	0.72	0.75	0.72	0.81	0.77	0.77	0.8
	DV	0.42	0.56	0.69	0.64	0.57	0.7	0.64	0.76	0.67	0.73
Glass1	CS	0.64	0.89	0.87	0.9	0.82	0.79	0.69	0.8	0.81	0.84
	CS wt/ SMOTE	0.6	0.66	0.77	0.78	0.78	0.81	0.79	0.77	0.81	0.79
	CV	0.6	0.71	0.84	0.77	0.8	0.8	0.71	0.82	0.84	0.78
	CV wt/ SMOTE	0.57	0.66	0.77	0.8	0.83	0.82	0.84	0.81	0.84	0.79
	Tradt	0.56	0.68	0.78	0.76	0.79	0.77	0.81	0.81	0.82	0.82
	Tradt wt/ SMOTE	0.59	0.71	0.82	0.77	0.77	0.79	0.75	0.86	0.84	0.84
	DS	0.39	0.39	0.8	0.74	0.79	0.68	0.77	0.8	0.75	0.77
	DV	0.5	0.7	0.7	0.68	0.8	0.78	0.75	0.79	0.77	0.74
New-Thyroid	CS	1.0	0.98	1.0	1.0	1.0	0.98	0.98	0.96	0.96	0.96
	CS wt/ SMOTE	0.88	0.78	0.97	1.0	0.98	0.98	0.98	0.98	0.98	1.0
	CV	1.0	0.98	0.98	0.98	1.0	1.0	1.0	1.0	1.0	1.0
	CV wt/ SMOTE	0.88	0.98	0.97	0.98	0.98	0.96	0.98	0.98	0.98	0.98
	Tradt	0.96	1.0	0.96	0.98	0.98	0.98	0.98	0.98	0.98	0.98
	Tradt wt/ SMOTE	0.95	0.96	0.98	1.0	0.98	0.96	0.96	0.96	0.96	0.98
	DS	0.96	0.85	0.96	0.98	0.98	0.96	0.98	0.98	0.98	1.0
	DV	0.81	0.79	0.85	0.85	0.96	0.94	0.96	0.94	0.96	0.96
Dermatology	CS	0.97	1.0	0.98	0.98	0.96	1.0	1.0	1.0	1.0	0.99
	CS wt/ SMOTE	0.82	0.86	0.89	0.92	0.88	0.96	0.94	0.97	0.95	0.96
	CV	1.0	1.0	1.0	1.0	0.99	0.99	0.99	0.99	0.99	0.99
	CV wt/ SMOTE	0.86	0.91	0.97	0.97	0.98	0.96	0.96	0.97	0.97	0.97
	Tradt	0.97	0.95	0.97	0.98	0.97	0.96	0.96	0.98	1.0	0.97
	Tradt wt/ SMOTE	0.95	0.97	0.97	0.98	0.98	0.97	0.99	0.99	0.98	0.98
	DS	0.91	0.88	0.91	0.93	0.92	0.92	0.94	0.92	0.93	0.95
	DV	0.94	0.96	0.96	0.96	0.98	0.95	0.94	0.98	0.95	0.98
Balance	CS	0.84	0.83	0.85	0.87	0.9	0.89	0.92	0.91	0.93	0.92
	CS wt/ SMOTE	0.66	0.6	0.77	0.86	0.88	0.89	0.9	0.91	0.86	0.87
	CV	0.78	0.75	0.74	0.72	0.71	0.76	0.72	0.8	0.73	0.77
	CV wt/ SMOTE	0.69	0.72	0.7	0.7	0.72	0.71	0.67	0.79	0.7	0.77
	Tradt	0.81	0.77	0.8	0.86	0.87	0.85	0.91	0.93	0.9	0.95
	Tradt wt/ SMOTE	0.8	0.77	0.88	0.93	0.94	0.9	0.88	0.89	0.95	0.92
	DS	0.68	0.73	0.83	0.84	0.9	0.86	0.88	0.92	0.92	0.93
	DV	0.38	0.46	0.48	0.55	0.62	0.62	0.61	0.78	0.69	0.69
Segment0	CS	0.99	0.99	0.99	0.99	0.99	1.0	1.0	1.0	1.0	1.0
	CS wt/ SMOTE	0.99	0.99	0.99	0.99	1.0	1.0	1.0	1.0	1.0	1.0
	CV	0.95	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.0	0.99
	CV wt/ SMOTE	0.95	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Tradt	0.98	0.99	0.99	0.99	0.99	1.0	0.99	1.0	1.0	1.0
	Tradt wt/ SMOTE	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	DS	0.92	0.97	0.98	0.99	0.99	0.99	0.98	1.0	0.99	0.99
	DV	0.97	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Table 3 (continued)

Database	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
DryBean	CS	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
	CS wt/ SMOTE	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
	CV	0.92	0.92	0.92	0.92	0.92	0.93	0.93	0.93	0.93	0.93
	CV wt/ SMOTE	0.92	0.91	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
	Tradt	0.92	0.93	0.93	0.93	0.93	0.93	0.94	0.93	0.93	0.94
	Tradt wt/ SMOTE	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
	DS	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
	DV	0.92	0.92	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93
Glass	CS	0.45	0.5	0.68	0.68	0.66	0.65	0.65	0.67	0.63	0.68
	CS wt/ SMOTE	0.34	0.37	0.57	0.53	0.59	0.58	0.6	0.61	0.58	0.63
	CV	0.58	0.6	0.76	0.66	0.7	0.67	0.63	0.63	0.66	0.73
	CV wt/ SMOTE	0.35	0.47	0.56	0.58	0.56	0.61	0.66	0.71	0.63	0.67
	Tradt	0.41	0.41	0.59	0.57	0.48	0.66	0.65	0.64	0.66	0.68
	Tradt wt/ SMOTE	0.41	0.4	0.49	0.54	0.56	0.6	0.6	0.65	0.66	0.73
	DS	0.29	0.31	0.51	0.48	0.51	0.61	0.63	0.65	0.74	0.65
	DV	0.3	0.37	0.51	0.44	0.51	0.63	0.68	0.6	0.63	0.61
Page-Blocks0	CS	0.9	0.91	0.93	0.94	0.94	0.92	0.9	0.92	0.93	0.93
	CS wt/ SMOTE	0.84	0.86	0.88	0.9	0.89	0.92	0.91	0.92	0.91	0.9
	CV	0.89	0.88	0.91	0.92	0.9	0.89	0.89	0.89	0.9	0.89
	CV wt/ SMOTE	0.82	0.82	0.84	0.86	0.84	0.83	0.84	0.81	0.84	0.83
	Tradt	0.86	0.9	0.91	0.92	0.92	0.93	0.93	0.92	0.93	0.93
	Tradt wt/ SMOTE	0.88	0.91	0.91	0.92	0.93	0.92	0.93	0.93	0.93	0.93
	DS	0.85	0.86	0.86	0.9	0.91	0.9	0.92	0.91	0.91	0.92
	DV	0.86	0.88	0.91	0.93	0.93	0.93	0.92	0.93	0.94	0.94
Vowel0	CS	0.85	0.85	0.91	0.96	1.0	1.0	1.0	1.0	1.0	1.0
	CS wt/ SMOTE	0.83	0.88	0.87	0.92	1.0	0.99	1.0	1.0	1.0	1.0
	CV	0.83	0.87	0.89	0.9	0.93	0.94	0.93	0.98	0.97	0.95
	CV wt/ SMOTE	0.76	0.83	0.85	0.86	0.92	0.91	0.91	0.93	0.93	0.93
	Tradt	0.83	0.85	0.87	0.94	0.97	1.0	1.0	1.0	0.98	1.0
	Tradt wt/ SMOTE	0.79	0.85	0.86	0.88	0.96	0.98	0.98	1.0	1.0	1.0
	DS	0.82	0.85	0.85	0.87	1.0	1.0	1.0	1.0	0.99	1.0
	DV	0.78	0.79	0.86	0.87	0.94	0.94	0.94	1.0	1.0	1.0
Yeast4	CS	0.49	0.68	0.77	0.75	0.65	0.67	0.74	0.7	0.74	0.72
	CS wt/ SMOTE	0.49	0.55	0.6	0.6	0.55	0.55	0.55	0.55	0.65	0.69
	CV	0.6	0.61	0.62	0.61	0.6	0.6	0.61	0.59	0.6	0.61
	CV wt/ SMOTE	0.6	0.59	0.65	0.58	0.59	0.57	0.59	0.57	0.57	0.6
	Tradt	0.57	0.59	0.75	0.66	0.72	0.63	0.7	0.69	0.65	0.66
	Tradt wt/ SMOTE	0.62	0.58	0.75	0.61	0.65	0.6	0.65	0.65	0.69	0.69
	DS	0.49	0.55	0.66	0.59	0.59	0.6	0.57	0.65	0.63	0.62
	DV	0.48	0.49	0.83	0.57	0.55	0.55	0.59	0.63	0.59	0.53
HCV	CS	0.49	0.65	0.76	0.49	0.83	0.69	0.69	0.79	0.83	0.78
	CS wt/ SMOTE	0.28	0.45	0.4	0.46	0.7	0.51	0.58	0.61	0.73	0.54
	CV	0.52	0.61	0.5	0.58	0.69	0.62	0.69	0.76	0.83	0.84
	CV wt/ SMOTE	0.42	0.5	0.46	0.48	0.64	0.5	0.6	0.69	0.75	0.74
	Tradt	0.43	0.45	0.54	0.55	0.5	0.59	0.66	0.75	0.68	0.68
	Tradt wt/ SMOTE	0.39	0.47	0.49	0.54	0.56	0.57	0.72	0.71	0.72	0.72
	DS	0.19	0.3	0.45	0.38	0.51	0.48	0.63	0.61	0.65	0.64
	DV	0.44	0.37	0.26	0.42	0.39	0.57	0.47	0.43	0.51	0.4

Table 3 (continued)

Database	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Yeast	CS	–	0.58	0.6	0.6	0.61	0.61	0.6	0.57	0.59	0.58
	CS wt/ SMOTE	–	0.56	0.54	0.55	0.56	0.52	0.51	0.52	0.56	0.58
	CV	–	0.37	0.46	0.49	0.46	0.44	0.48	0.42	0.4	0.44
	CV wt/ SMOTE	–	0.35	0.4	0.34	0.33	0.37	0.39	0.37	0.35	0.38
	Tradt	–	0.56	0.59	0.6	0.61	0.61	0.59	0.58	0.61	0.56
	Tradt wt/ SMOTE	–	0.58	0.57	0.62	0.62	0.61	0.58	0.57	0.59	0.57
	DS	–	0.5	0.54	0.5	0.58	0.58	0.58	0.57	0.6	0.55
	DV	–	0.42	0.43	0.42	0.46	0.49	0.5	0.49	0.47	0.5

Table 4 Silhouette coefficient

Database	Without SMOTE	With SMOTE
Hayes-Roth	0.02 (0.03)	0.09 (0.04)
Glass1	0.08 (0.04)	0.11 (0.06)
New-Thyroid	0.57 (0.06)	0.60 (0.02)
Dermatology	0.39 (0.04)	0.43 (0.02)
Balance	0.08 (0.05)	0.12 (0.02)
Segment0	0.71 (0.03)	0.72 (0.03)
DryBean	0.31 (0.01)	0.31 (0.01)
Glass	− 0.06 (0.04)	− 0.03 (0.02)
Page-Blocks0	0.44 (0.03)	0.43 (0.05)
Vowel0	0.51 (0.08)	0.54 (0.10)
Yeast4	0.25 (0.03)	0.41 (0.02)
HCV	0.25 (0.06)	0.26 (0.02)
Yeast	− 0.02 (0.01)	0.01 (0.01)

These outcomes emphasize the benefits of adopting metric learning to create task-specific dissimilarity measures, marking a significant shift from the conventional reliance on static distance functions.

5.2 Augmentation protocol

Our training protocol applied SMOTE to mitigate class imbalance challenges, particularly in the initial folds where the availability of training data is considerably constrained. To comprehensively assess the impact of utilizing this approach, we evaluated the contrastive dissimilarity space and vector models on both the original and SMOTE-augmented training sets.

Utilizing the unmodified dataset, the contrastive dissimilarity space model markedly declines, especially in the initial folds, with 8 wins (6.2%), 35 draws (26.9%), and 87 losses (66.9%), underscoring a significant difference ($t(129) = -9.3$, $p < .001$). The vector model, using the original dataset, also shows a decrease in effectiveness, particularly in the early folds, achieving 9 wins (6.9%), 20 draws (15.4%), and 101 losses (77.7%), indicating a significant difference ($t(129) = -8.61$, $p < .001$).

The Silhouette coefficient is a metric used to assess the quality of clustering by measuring both the separation between clusters and their internal cohesion. It ranges from -1 to 1, with values near 1 indicating well-separated and distinct clusters, and values closer to -1 suggesting clusters that are indistinct or overlapping. Table 4 showcases the Silhouette coefficient of the contrastive dissimilarity space matrix, comparing the impact of training the model with and without the SMOTE. The inclusion of SMOTE marginally improved the Silhouette coefficient, denoting better cluster distinction and cohesion, the improvement is statistically significant ($t(12) = -3.1$, $p = 0.01$).

5.3 Balanced databases

We extended our analysis to more balanced datasets to assess the efficacy of our proposed method. Specifically, we employed 6 databases from UC Irvine Machine Learning Repository [18] and one from Keel repository [2]. We targeted primarily on datasets with an imbalance ratio nearing one, indicating a near-equal distribution between the majority and minority classes. Table 5 presents the datasets with their main characteristics.

The contrastive dissimilarity space model shows a slight edge over traditional models with SMOTE, achieving 33 wins (47.1%), 15 draws (21.4%), and 22 losses (31.4%). Statistical analysis yields a t value of -1.22 and a p value of 0.227. Against traditional models without SMOTE, it records 32 wins (45.7%), 14 draws (20%), and 24 losses (34.3%), with a t value of -1.69 and a p value of 0.096. The effectiveness becomes more pronounced when training data is limited (only 30% of the dataset, covering the first

Table 5 Balanced databases

No	Source	Database	Instances	Features	Classes	Imbalance ratio
01	UCI	Raisin [7]	900	7	2	1.00
02	UCI	TME [12]	400	50	4	1.00
03	UCI	Sirtuin6 [43]	100	6	2	1.00
04	Keel	Penbased	1100	16	10	1.10
05	UCI	Coimbra [26]	116	9	2	1.23
06	UCI	Algerian Fire [50]	244	10	2	1.30
07	UCI	TCGA-LGG [44]	839	23	2	1.38

Table 6 Balanced data F1-score

Database	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Raisin	CS	0.88	0.88	0.87	0.89	0.88	0.9	0.9	0.88	0.89	0.88
	CV	0.84	0.87	0.85	0.87	0.87	0.88	0.88	0.88	0.88	0.88
	Tradt	0.84	0.88	0.85	0.88	0.88	0.87	0.88	0.89	0.88	0.87
	Tradt w/ SMOTE	0.84	0.88	0.85	0.88	0.88	0.87	0.87	0.89	0.88	0.87
TME	CS	0.43	0.53	0.63	0.65	0.6	0.67	0.62	0.68	0.69	0.74
	CV	0.44	0.58	0.63	0.66	0.69	0.7	0.73	0.71	0.72	0.71
	Tradt	0.64	0.69	0.69	0.73	0.7	0.72	0.7	0.72	0.72	0.72
	Tradt w/ SMOTE	0.64	0.71	0.72	0.71	0.73	0.7	0.73	0.72	0.77	0.72
Sirtuin6	CS	0.87	0.93	0.77	0.8	0.86	0.8	0.76	0.69	0.7	0.7
	CV	0.86	0.93	0.7	0.8	0.77	0.73	0.8	0.76	0.7	0.76
	Tradt	0.87	0.83	0.87	0.8	0.8	0.87	0.93	0.83	0.9	0.83
	Tradt w/ SMOTE	0.8	0.86	0.9	0.8	0.83	0.83	0.93	0.83	0.87	0.83
Penbased	CS	0.95	0.98	0.99	1.0	0.99	0.98	0.99	0.99	0.99	0.98
	CV	0.86	0.87	0.91	0.91	0.91	0.92	0.93	0.93	0.94	0.93
	Tradt	0.9	0.93	0.97	0.96	0.97	0.99	0.99	0.99	0.98	0.98
	Tradt w/ SMOTE	0.9	0.93	0.97	0.96	0.98	0.99	0.99	0.99	0.98	0.98
Coimbra	CS	0.6	0.88	0.8	0.77	0.82	0.71	0.76	0.66	0.71	0.7
	CV	0.57	0.82	0.77	0.77	0.79	0.71	0.77	0.6	0.68	0.74
	Tradt	0.69	0.7	0.74	0.73	0.7	0.79	0.76	0.65	0.68	0.74
	Tradt w/ SMOTE	0.66	0.74	0.76	0.74	0.76	0.79	0.79	0.68	0.74	0.74
Algerian Fire	CS	0.94	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	CV	0.99	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Tradt	0.9	1.0	0.99	1.0	1.0	1.0	1.0	0.97	0.97	0.99
	Tradt w/ SMOTE	0.87	1.0	0.99	1.0	1.0	1.0	1.0	0.96	0.97	0.99
TCGA-LGG	CS	0.88	0.88	0.85	0.85	0.85	0.83	0.84	0.83	0.84	0.82
	CV	0.87	0.87	0.86	0.86	0.85	0.84	0.84	0.84	0.84	0.84
	Tradt	0.84	0.85	0.84	0.85	0.84	0.84	0.83	0.83	0.83	0.83
	Tradt w/ SMOTE	0.84	0.85	0.85	0.84	0.84	0.83	0.83	0.83	0.83	0.83

three folds), it records 13 wins (61.9%), 3 draws (14.3%), and 5 losses (23.8%) in both scenarios ($t(20) = 0.08$, $p = 0.940$). Table 6 showcases the results.

Similarly, the contrastive dissimilarity vector model underperforms when compared to traditional models with SMOTE, securing 23 wins (32.9%), 14 draws (20%), and 33 losses (47.1%), evidenced by a t value of -2.79 and a p value of 0.007. When facing traditional models without SMOTE, it achieves 24 wins (34.3%), 12 draws (17.1%), and 34 losses (48.6%), with a t value of -3.37 and a p value of 0.001. The approach shows a slight improvement in performance with limited training data, but not to a significant extent.

These findings point to some advantages of the contrastive dissimilarity model, particularly in scenarios with restricted training data. Nonetheless, the statistical significance of these advantages remains low, highlighting the need for additional studies in balanced datasets.

5.4 Computational complexity

Computational complexity can be divided into three main steps: the input size, the neural network, and the dissimilarity representation. The input size depends exclusively on the data being evaluated, and thus, we cannot make any strong assumptions about it. We denote its computational complexity as $O(\text{input})$.

The neural network training computational complexity is primarily determined by the projection head architecture, meaning that the more parameters it has, the more time it will take to train and predict. In this work, the larger evaluated projection head contained around six hundred thousand parameters. We shall denote its training computational complexity as $O(\text{network}_{\text{train}})$.

The final step is the dissimilarity representation, which has two approaches: space and vector. The dissimilarity space builds a matrix with n rows and m columns, where n represents the number of samples and m the number of prototypes, resulting in a complexity of $O(n \cdot m \cdot \text{network}_{\text{predict}})$. The dissimilarity vector composes a feature vector for each combination of sample and prototype, generating w dissimilarity values for each, resulting in a complexity of $O(n \cdot m \cdot \text{network}_{\text{predict}} \cdot w)$.

6 Conclusion

This study introduced a novel approach to classifying imbalanced tabular datasets that combines metric learning and dissimilarity, known as contrastive dissimilarity. Our proposal improved robustness and classification performance by developing a task-specific dissimilarity function that went beyond traditional distance measures.

Experiments were conducted in a comprehensive experimental setting with varying training sizes. The contrastive dissimilarity space approach was especially useful in scenarios with limited data availability. It consistently outperformed traditional machine learning algorithms, including SVM, random forest, and Naive Bayes, with over 90% of wins and ties. While its performance in balanced dataset contexts was variable, it still showed promising results, particularly with limited training data.

While our approach produced promising results that warrant further investigation, it has two significant practical limitations: First, the data availability threshold varies depending on the scenario being evaluated, making it difficult to pinpoint a precise threshold at which our method outperforms traditional approaches. Second, our proposal includes some hyperparameters that need to be fine-tuned for optimal performance, such as the projection head architecture, temperature, learning rate, batch size, prototype selection, and the use of Synthetic Minority Oversampling Technique (SMOTE).

Finally, our proposed method represents a promising advance in handling complex and imbalanced datasets. Its flexibility and robust performance under changing conditions make it useful for a wide range of applications. Potential applications include domains that deal with imbalanced data, such as medical diagnoses, fraud detection, and rare events in general.

Future work will focus on improving the dissimilarity vector model and determining the data availability threshold. Furthermore, we intend to test additional long-tail classification scenarios using image datasets such as CIFAR10-LT, CIFAR100-LT, and ImageNet-LT.

Appendix A Traditional models

This appendix presents Table 7 containing performance of three traditional machine learning models with and without SMOTE applied to the datasets in this study: support vector machine (SVM), random forest, and Naive Bayes. The performance of each model is evaluated using the F1-score

Table 7 Traditional models F1-score

Database	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Hayes-Roth	SVM	0.49	0.59	0.56	0.58	0.7	0.72	0.72	0.78	0.77	0.79
	SVM with SMOTE	0.51	0.57	0.45	0.58	0.72	0.72	0.72	0.78	0.77	0.8
	RF	0.33	0.54	0.6	0.55	0.68	0.75	0.75	0.77	0.77	0.77
	RF with SMOTE	0.46	0.49	0.57	0.56	0.67	0.74	0.78	0.77	0.82	0.77
	NB	0.49	0.49	0.46	0.51	0.56	0.59	0.62	0.68	0.69	0.62
	NB with SMOTE	0.49	0.55	0.46	0.46	0.58	0.56	0.62	0.67	0.69	0.62
Glass1	SVM	0.42	0.71	0.76	0.76	0.77	0.75	0.75	0.75	0.74	0.73
	SVM with SMOTE	0.51	0.68	0.78	0.7	0.79	0.77	0.78	0.73	0.75	0.75
	RF	0.59	0.68	0.82	0.77	0.75	0.79	0.73	0.86	0.84	0.84
	RF with SMOTE	0.56	0.65	0.75	0.76	0.74	0.76	0.81	0.81	0.82	0.82
	NB	0.48	0.6	0.6	0.61	0.56	0.53	0.56	0.58	0.62	0.6
	NB with SMOTE	0.49	0.58	0.6	0.58	0.58	0.58	0.6	0.63	0.62	0.6
New-Thyroid	SVM	0.95	0.95	0.98	1.0	0.98	0.96	0.96	0.96	0.96	0.98
	SVM with SMOTE	0.53	1.0	0.87	0.9	0.98	0.96	0.96	0.98	0.96	0.94
	RF	0.91	0.96	0.96	0.9	0.92	0.9	0.92	0.92	0.9	0.94
	RF with SMOTE	0.96	0.94	0.92	0.94	0.92	0.94	0.96	0.94	0.94	0.96
	NB	0.91	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96
	NB with SMOTE	0.85	0.96	0.96	0.98	0.98	0.98	0.98	0.96	0.98	0.98
Glass1	SVM	0.95	0.97	0.95	0.96	0.92	0.97	0.94	0.97	0.96	0.97
	SVM with SMOTE	0.92	0.94	0.74	0.97	0.93	0.96	0.94	0.97	0.95	0.97
	RF	0.89	0.94	0.97	0.98	0.98	0.97	0.99	0.99	0.98	0.98
	RF with SMOTE	0.97	0.95	0.97	0.98	0.97	0.95	0.96	0.98	1.0	0.97
	NB	0.88	0.94	0.84	0.88	0.82	0.82	0.82	0.82	0.82	0.82
	NB with SMOTE	0.9	0.95	0.84	0.9	0.82	0.82	0.82	0.82	0.82	0.82
Balance	SVM	0.8	0.77	0.88	0.93	0.94	0.9	0.88	0.89	0.95	0.92
	SVM with SMOTE	0.81	0.77	0.8	0.86	0.87	0.85	0.91	0.93	0.9	0.95
	RF	0.52	0.56	0.57	0.57	0.6	0.61	0.6	0.6	0.6	0.61
	RF with SMOTE	0.51	0.57	0.57	0.59	0.59	0.6	0.6	0.6	0.59	0.62
	NB	0.62	0.58	0.55	0.63	0.61	0.6	0.62	0.62	0.62	0.63
	NB with SMOTE	0.61	0.57	0.6	0.62	0.66	0.64	0.65	0.67	0.69	0.67
Segment0	SVM	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	SVM with SMOTE	0.92	0.99	0.99	0.99	0.99	1.0	0.99	1.0	1.0	1.0
	RF	0.98	0.98	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99
	RF with SMOTE	0.98	0.98	0.98	0.98	0.99	0.98	0.99	0.99	0.99	0.99
	NB	0.7	0.7	0.68	0.69	0.69	0.75	0.76	0.76	0.77	0.78
	NB with SMOTE	0.72	0.71	0.69	0.69	0.7	0.76	0.76	0.76	0.76	0.78
DryBean	SVM	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
	SVM with SMOTE	0.92	0.89	0.9	0.91	0.91	0.92	0.94	0.92	0.93	0.92
	RF	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.94	0.94
	RF with SMOTE	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.94
	NB	0.9	0.9	0.9	0.9	0.9	0.9	0.91	0.91	0.91	0.91
	NB with SMOTE	0.9	0.9	0.9	0.9	0.91	0.91	0.91	0.91	0.91	0.91
Glass	SVM	0.41	0.4	0.49	0.54	0.56	0.6	0.6	0.65	0.66	0.64
	SVM with SMOTE	0.16	0.38	0.59	0.55	0.47	0.66	0.65	0.64	0.65	0.61
	RF	0.41	0.39	0.47	0.47	0.42	0.51	0.56	0.62	0.66	0.73
	RF with SMOTE	0.41	0.41	0.55	0.57	0.48	0.54	0.61	0.62	0.66	0.68
	NB	0.29	0.29	0.45	0.4	0.47	0.48	0.47	0.46	0.46	0.46
	NB with SMOTE	0.29	0.26	0.36	0.35	0.45	0.52	0.55	0.51	0.54	0.52

Table 7 (continued)

Database	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Page-Blocks0	SVM	0.86	0.89	0.9	0.92	0.91	0.92	0.93	0.93	0.92	0.92
	SVM with SMOTE	0.84	0.87	0.86	0.89	0.88	0.9	0.91	0.91	0.89	0.91
	RF	0.88	0.91	0.91	0.91	0.93	0.92	0.92	0.93	0.93	0.93
	RF with SMOTE	0.86	0.9	0.91	0.92	0.92	0.93	0.93	0.92	0.93	0.93
	NB	0.69	0.71	0.75	0.76	0.76	0.75	0.75	0.73	0.72	0.72
	NB with SMOTE	0.76	0.75	0.76	0.76	0.77	0.76	0.76	0.75	0.75	0.73
Vowel0	SVM	0.78	0.83	0.82	0.8	0.96	0.98	0.98	1.0	1.0	1.0
	SVM with SMOTE	0.83	0.85	0.85	0.92	0.97	1.0	1.0	1.0	0.94	1.0
	RF	0.77	0.85	0.86	0.88	0.94	0.94	0.96	0.96	0.96	0.98
	RF with SMOTE	0.76	0.84	0.87	0.94	0.96	0.96	0.96	0.98	0.98	0.98
	NB	0.79	0.79	0.8	0.79	0.8	0.8	0.81	0.83	0.8	0.82
	NB with SMOTE	0.72	0.75	0.8	0.79	0.74	0.77	0.78	0.77	0.77	0.78
Yeast4	SVM	0.62	0.58	0.75	0.61	0.65	0.6	0.65	0.65	0.69	0.69
	SVM with SMOTE	0.57	0.59	0.73	0.59	0.57	0.58	0.61	0.52	0.62	0.6
	RF	0.49	0.49	0.55	0.49	0.55	0.55	0.55	0.55	0.61	0.61
	RF with SMOTE	0.49	0.52	0.75	0.66	0.72	0.63	0.7	0.69	0.65	0.66
	NB	0.19	0.07	0.09	0.09	0.1	0.11	0.13	0.13	0.13	0.15
	NB with SMOTE	0.25	0.09	0.14	0.13	0.15	0.15	0.14	0.16	0.15	0.18
HCV	SVM	0.39	0.29	0.47	0.34	0.56	0.57	0.57	0.64	0.6	0.57
	SVM with SMOTE	0.26	0.37	0.35	0.39	0.41	0.56	0.46	0.41	0.5	0.67
	RF	0.36	0.47	0.48	0.45	0.44	0.36	0.44	0.5	0.49	0.52
	RF with SMOTE	0.43	0.45	0.54	0.49	0.5	0.59	0.56	0.75	0.68	0.68
	NB	0.28	0.33	0.49	0.54	0.49	0.53	0.72	0.71	0.72	0.72
	NB with SMOTE	0.28	0.33	0.44	0.55	0.46	0.55	0.66	0.65	0.67	0.67
Yeast	SVM	–	0.58	0.53	0.62	0.62	0.58	0.58	0.56	0.56	0.57
	SVM with SMOTE	–	0.29	0.33	0.34	0.31	0.35	0.34	0.39	0.44	0.42
	RF	–	0.39	0.57	0.49	0.62	0.61	0.57	0.57	0.59	0.51
	RF with SMOTE	–	0.56	0.59	0.6	0.61	0.61	0.59	0.58	0.61	0.56
	NB	–	0.32	0.44	0.34	0.34	0.32	0.32	0.33	0.33	0.33
	NB with SMOTE	–	0.32	0.39	0.34	0.31	0.33	0.33	0.38	0.36	0.37
Raisin	SVM	0.82	0.88	0.85	0.88	0.88	0.87	0.87	0.89	0.88	0.87
	SVM with SMOTE	0.82	0.88	0.85	0.88	0.88	0.87	0.87	0.89	0.88	0.87
	RF	0.82	0.88	0.85	0.88	0.88	0.87	0.87	0.89	0.88	0.87
	RF with SMOTE	0.84	0.84	0.81	0.86	0.86	0.86	0.86	0.86	0.88	0.87
	NB	0.82	0.84	0.82	0.82	0.84	0.84	0.82	0.82	0.82	0.82
	NB with SMOTE	0.84	0.84	0.81	0.86	0.86	0.86	0.86	0.86	0.88	0.87
TME	SVM	0.84	0.84	0.81	0.86	0.86	0.86	0.86	0.86	0.88	0.87
	SVM with SMOTE	0.64	0.69	0.69	0.73	0.7	0.72	0.7	0.72	0.7	0.71
	RF	0.64	0.69	0.69	0.73	0.7	0.72	0.7	0.72	0.7	0.71
	RF with SMOTE	0.64	0.69	0.69	0.73	0.7	0.72	0.7	0.72	0.7	0.71
	NB	0.64	0.69	0.69	0.73	0.7	0.72	0.7	0.72	0.7	0.71
	NB with SMOTE	0.59	0.63	0.65	0.69	0.66	0.68	0.64	0.65	0.72	0.72
Sirtuin6	SVM	0.87	0.79	0.8	0.73	0.72	0.83	0.93	0.83	0.8	0.7
	SVM with SMOTE	0.8	0.79	0.77	0.73	0.71	0.83	0.93	0.83	0.8	0.7
	RF	0.87	0.79	0.8	0.73	0.72	0.83	0.93	0.83	0.8	0.7
	RF with SMOTE	0.8	0.79	0.77	0.73	0.71	0.83	0.93	0.83	0.8	0.7
	NB	0.87	0.79	0.8	0.73	0.72	0.83	0.93	0.83	0.8	0.7
	NB with SMOTE	0.52	0.72	0.83	0.8	0.73	0.73	0.77	0.77	0.73	0.73

Table 7 (continued)

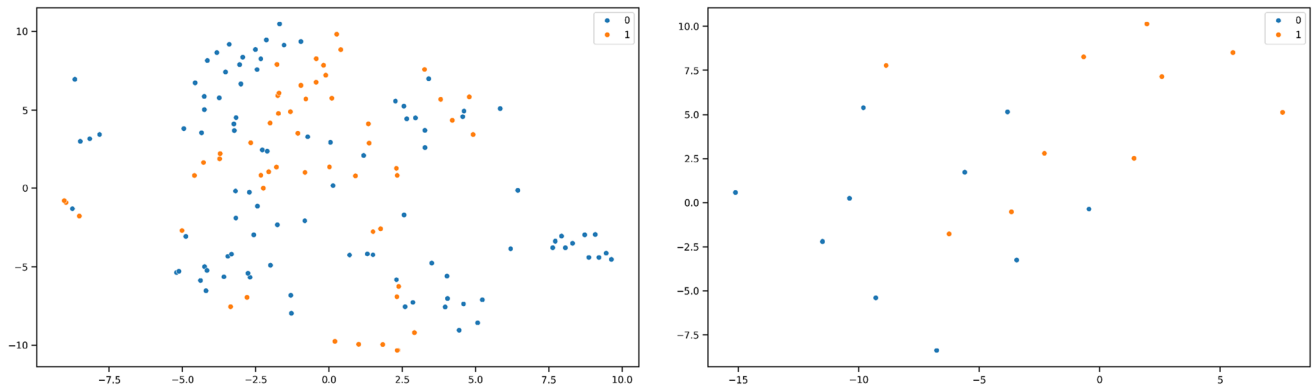
Database	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Penbased	SVM	0.9	0.93	0.97	0.96	0.97	0.99	0.99	0.99	0.98	0.98
	SVM with SMOTE	0.75	0.79	0.8	0.82	0.82	0.82	0.82	0.81	0.82	0.82
	RF	0.84	0.9	0.92	0.91	0.93	0.95	0.95	0.95	0.96	0.96
	RF with SMOTE	0.81	0.9	0.91	0.93	0.93	0.95	0.95	0.96	0.96	0.96
	NB	0.75	0.79	0.8	0.82	0.82	0.82	0.82	0.81	0.82	0.82
	NB with SMOTE	0.81	0.9	0.91	0.93	0.93	0.95	0.95	0.96	0.96	0.96
Coimbra	SVM	0.63	0.7	0.74	0.67	0.7	0.74	0.74	0.57	0.68	0.74
	SVM with SMOTE	0.66	0.63	0.68	0.68	0.68	0.71	0.65	0.62	0.68	0.68
	RF	0.69	0.63	0.71	0.73	0.7	0.79	0.76	0.65	0.68	0.68
	RF with SMOTE	0.57	0.68	0.76	0.73	0.73	0.79	0.79	0.62	0.68	0.74
	NB	0.66	0.63	0.68	0.68	0.68	0.71	0.65	0.62	0.68	0.68
	NB with SMOTE	0.63	0.63	0.63	0.66	0.63	0.71	0.65	0.62	0.65	0.68
Algerian Fire	SVM	0.86	0.92	0.96	0.93	0.97	0.97	0.96	0.92	0.93	0.94
	SVM with SMOTE	0.87	0.92	0.9	0.9	0.92	0.92	0.92	0.92	0.92	0.93
	RF	0.9	1.0	0.99	1.0	1.0	1.0	1.0	0.97	0.97	0.99
	RF with SMOTE	0.87	0.92	0.9	0.9	0.92	0.92	0.92	0.92	0.92	0.93
	NB	0.87	0.92	0.9	0.9	0.92	0.92	0.92	0.92	0.92	0.93
	NB with SMOTE	0.87	0.92	0.9	0.9	0.92	0.92	0.92	0.92	0.92	0.93
TCGA-LGG	SVM	0.87	0.92	0.9	0.9	0.92	0.92	0.92	0.92	0.92	0.93
	SVM with SMOTE	0.84	0.85	0.85	0.84	0.83	0.83	0.83	0.83	0.83	0.83
	RF	0.83	0.83	0.81	0.81	0.83	0.84	0.83	0.82	0.81	0.81
	RF with SMOTE	0.82	0.83	0.84	0.83	0.84	0.82	0.82	0.81	0.81	0.82
	NB	0.6	0.55	0.56	0.55	0.57	0.74	0.72	0.72	0.72	0.7
	NB with SMOTE	0.64	0.57	0.57	0.58	0.58	0.77	0.71	0.7	0.76	0.71

metric. The results are organized in a tabular format, with the datasets listed in rows and varying proportions of training data, ranging from 10 to 100%, displayed in columns.

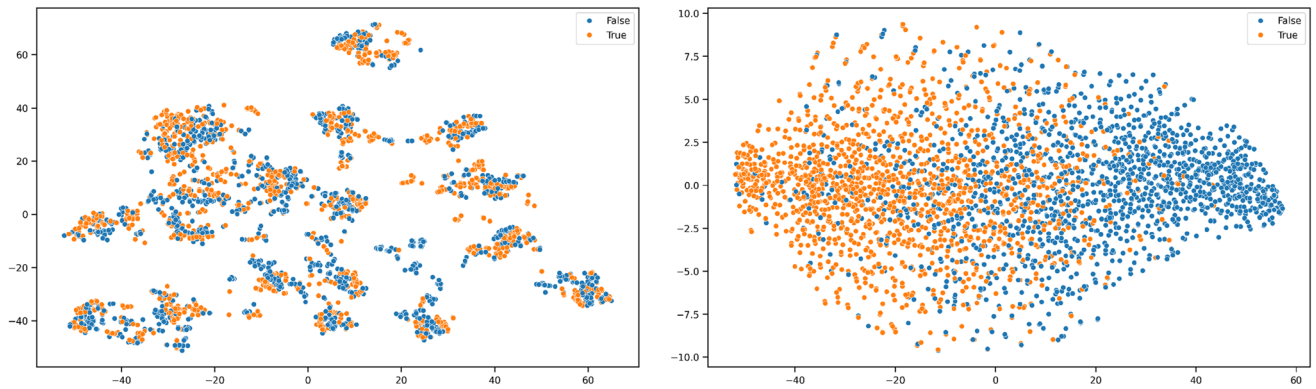
Appendix B Dissimilarity vector

The performance of the dissimilarity vector was lower than anticipated, prompting us to detail its shortcomings in this appendix. The dissimilarity vector has a very simplistic nature that rely on the differences between the representations of samples. In tabular data, these representations are relatively fixed, as a result, when the data points are not distinctly separate, the dissimilarity vector tends to reflect less discriminative differences. Figure 2 illustrates this concept using a sample fold from the Glass1 dataset, the lack of distinctiveness in the original features, and thus in the prototypes, leads to a untidy dissimilarity representation. For ease visualization, the data is condensed into two dimensions using t-SNE.

Contrastingly, Fig. 3 depicts the dissimilarity space representation, noticeably more structured and coherent, particularly in our proposed method.

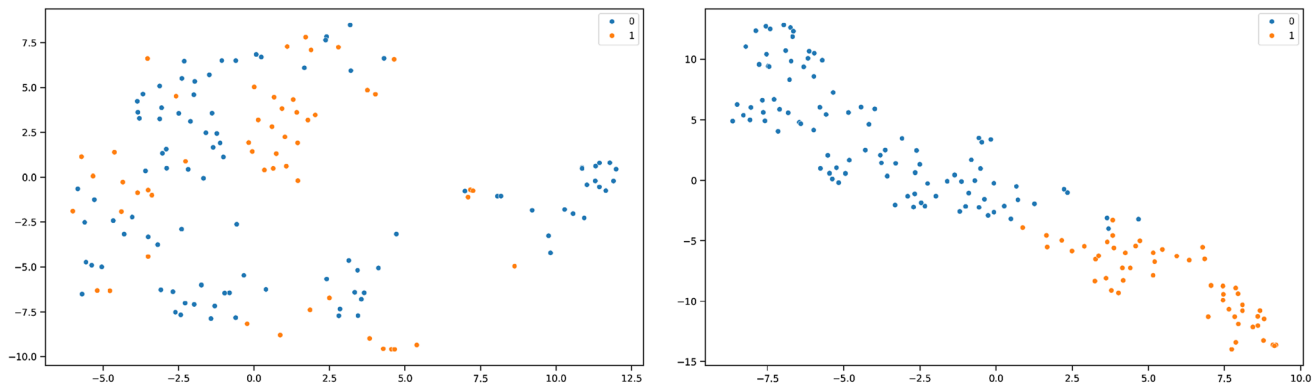


(a) Data. (b) Prototypes.



(c) Traditional dissimilarity vector. (d) Contrastive dissimilarity vector.

Fig. 2 Glass1 dissimilarity vector



(a) Traditional. (b) Contrastive.

Fig. 3 Glass1 dissimilarity space

Author contributions All authors contributed to the study conception and design. Data preparation and analysis were performed by Lucas Teixeira. The first draft of the manuscript was written by Lucas Teixeira, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and also supported by the Grant STICAmSud 23-STIC-13. Additional funding was provided by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil (CNPq) through Grants 101842/2024-4, 406030/2023-5, and 441610/2023-4.

Availability of data and material The data utilized in this paper are accessible through the following links: 1. UCI Machine Learning Repository: https://archive.ics.uci.edu/2. KEEL Repository: http://www.keel.es/The code utilized in this study is publicly available on GitHub at the following repository: https://github.com/lucasxteixeira/contrastive-dissimilarity.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Agrawal A (2019) Dissimilarity learning via siamese network predicts brain imaging data. arXiv preprint arXiv:1907.02591
2. Alcalá-Fdez J, Fernandez A, Luengo J et al (2011) Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J Multiple-Valued Logic Soft Comput 17(2–3):255–287
3. Cha SH, Srihari SN (2000) (2000) Writer identification: statistical analysis and dichotomizer. Advances in Pattern Recognition: Joint IAPR International Workshops SSPR 2000 and SPR 2000 Alicante, Spain, August 30-September 1. Proceedings, Springer, pp 123–132
4. Chawla NV, Bowyer KW, Hall LO et al (2002) Smote: Synthetic minority over-sampling technique. J Artif Intell Res 16:321–357. https://doi.org/10.1613/jair.953
5. Chen T, Kornblith S, Norouzi M, et al (2020) A simple framework for contrastive learning of visual representations. In: III HD, Singh A (eds) Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 119. PMLR, pp 1597–1607, https://proceedings.mlr.press/v119/chen20j.html
6. Chen Y, Hu Y, Hu X et al (2022) Cogo: a contrastive learning framework to predict disease similarity based on gene network and ontology structure. Bioinformatics 38(18):4380–4386. https://doi.org/10.1093/bioinformatics/btac520
7. Çinar İ, Koklu M, TaŞdemİr Ş (2020) Kuru Üzüm tanelerinin makine görüşü ve yapay zeka yöntemleri kullanılarak sınıflandırılması. Gazi Journal of Engineering Sciences 6(3), 200–209. https://doi.org/10.30855/gmbd.2020.03.03
8. Cocos A, Qian T, Callison-Burch C et al (2017) Crowd control: effectively utilizing unscreened crowd workers for biomedical data annotation. J Biomed Inform 69:86–92. https://doi.org/10.1016/j.jbi.2017.04.003
9. Costa YMG, Bertolini D, Britto AS et al (2019) The dissimilarity approach: a review. Artif Intell Rev 53(4):2783–2808. https://doi.org/10.1007/s10462-019-09746-z
10. Dosovitskiy A, Springenberg JT, Riedmiller M et al (2014) Discriminative unsupervised feature learning with convolutional neural networks. In: Ghahramani Z, Welling M, Cortes C et al (eds) Advances in neural information processing systems, vol 27. Curran Associates Inc, New York
11. Duin RP, Pękalska E (2012) The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recogn Lett 33(7):826–832. https://doi.org/10.1016/j.patrec.2011.04.019
12. Er MB, Aydilek IB (2019) Music emotion recognition by using chroma spectrogram and deep visual features. Int J Comput Intell Syst 12(2):1622. https://doi.org/10.2991/ijcis.d.191216.001
13. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Balcan MF, Weinberger KQ (eds) Proceedings of The 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 48. PMLR, New York, New York, USA, pp 1050–1059, https://proceedings.mlr.press/v48/gal16.html
14. Gharibshah Z, Zhu X (2022) Local contrastive feature learning for tabular data. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. Association for Computing Machinery, New York, NY, USA, CIKM '22, p 3963-3967, https://doi.org/10.1145/3511808.3557630
15. Hadsell R, Chopra S, LeCun Y (2006) Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06). IEEE, https://doi.org/10.1109/cvpr.2006.100
16. Hoffmann G, Bietenbeck A, Lichtinghagen R, et al (2018) Using machine learning techniques to generate laboratory diagnostic pathways-a case study. Journal of Laboratory and Precision Medicine 3(6). https://jlpm.amegroups.org/article/view/4401
17. Johnson JM, Khoshgoftaar TM (2019) Survey on deep learning with class imbalance. J Big Data. https://doi.org/10.1186/s40537-019-0192-5
18. Kelly M, Longjohn R, Nottingham K (2023) The uci machine learning repository. https://archive.ics.uci.edu
19. Khosla P, Teterwak P, Wang C et al (2020) Supervised contrastive learning. In: Larochelle H, Ranzato M, Hadsell R et al (eds) Advances in neural information processing systems, vol 33. Curran Associates Inc, New York, pp 18661–18673
20. Koklu M, Ozkan IA (2020) Multiclass classification of dry beans using computer vision and machine learning techniques. Comput Electron Agric 174:105507. https://doi.org/10.1016/j.compag.2020.105507
21. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. Progress Artif Intell 5(4):221–232. https://doi.org/10.1007/s13748-016-0094-0
22. Kulis B (2013) Metric learning: a survey. Found Trends Machine Learning 5(4):287–364. https://doi.org/10.1561/2200000019
23. Li T, Cao P, Yuan Y, et al (2022) Targeted supervised contrastive learning for long-tailed recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 6918–6928
24. Marrakchi Y, Makansi O, Brox T (2021) Fighting class imbalance with contrastive learning. Springer, Berlin, pp 466–476. https://doi.org/10.1007/978-3-030-87199-4_44
25. Mekhazni D, Bhuiyan A, Ekladious G, et al (2020) Unsupervised domain adaptation in the dissimilarity space for person re-identification. In: Computer Vision – ECCV 2020. Springer International Publishing, p 159–174,https://doi.org/10.1007/978-3-030-58583-9_10
26. Miguel Patrcio JP (2018) Breast cancer coimbra.https://doi.org/10.24432/C52P59, https://archive.ics.uci.edu/dataset/451

27. Nanni L, Brahnam S, Lumini A et al (2020) Animal sound classification using dissimilarity spaces. Appl Sci 10(23):8578. https://doi.org/10.3390/app10238578
28. Nanni L, Rigo A, Lumini A et al (2020) Spectrogram classification using dissimilarity space. Appl Sci 10(12):4176. https://doi.org/10.3390/app10124176
29. Nanni L, Minchio G, Brahnam S et al (2021) Experiments of image classification using dissimilarity spaces built with siamese networks. Sensors 21(5):1573. https://doi.org/10.3390/s21051573
30. Nanni L, Minchio G, Brahnam S et al (2021) Closing the performance gap between siamese networks for dissimilarity image classification and convolutional neural networks. Sensors 21(17):5809. https://doi.org/10.3390/s21175809
31. Nguyen GP, Worring M, Smeulders AWM (2006) Similarity learning via dissimilarity space in cbir. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval. Association for Computing Machinery, New York, NY, USA, MIR '06, p 107-116, https://doi.org/10.1145/1178677.1178695
32. Oord Avd, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. https://doi.org/10.48550/ARXIV.1807.03748, https://arxiv.org/abs/1807.03748
33. Orozco-Alzate M, Duin RP, Castellanos-Domínguez G (2009) A generalization of dissimilarity representations using feature lines and feature planes. Pattern Recogn Lett 30(3):242–25. https://doi.org/10.1016/j.patrec.2008.09.010
34. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
35. Pękalska E (2005) The dissimilarity representations in pattern recognition. concepts, theory and applications. PhD thesis, The University of Manchester
36. Pekalska E, Duin R (2006) Dissimilarity-based classification for vectorial representations. In: 18th International Conference on Pattern Recognition (ICPR'06). IEEE, https://doi.org/10.1109/icpr.2006.457
37. Pękalska E, Duin RP (2002) Dissimilarity representations allow for building good classifiers. Pattern Recogn Lett 23(8):943–956. https://doi.org/10.1016/s0167-8655(02)00024-7
38. Pekalska E, Paclik P, Duin RPW (2002) A generalized kernel approach to dissimilarity-based classification. J Mach Learn Res 2:175–211
39. Pinheiro RH, Cavalcanti GD, Tsang IR (2017) Combining dissimilarity spaces for text categorization. Inf Sci 406–407:87–101. https://doi.org/10.1016/j.ins.2017.04.025
40. Ruiz-Muñoz JF, Castellanos-Dominguez G, Orozco-Alzate M (2016) Enhancing the dissimilarity-based classification of birdsong recordings. Eco Inform 33:75–84. https://doi.org/10.1016/j.ecoinf.2016.04.001
41. Somorjai R, Dolenko B, Nikulin A et al (2011) Class proximity measures – dissimilarity-based classification and display of high-dimensional data. J Biomed Inform 44(5):775–788. https://doi.org/10.1016/j.jbi.2011.04.004
42. Souza VLF, Oliveira ALI, Sabourin R (2018) A writer-independent approach for offline signature verification using deep convolutional neural networks features. In: 2018 7th Brazilian Conference on Intelligent Systems (BRACIS). IEEE, https://doi.org/10.1109/bracis.2018.00044
43. Tardu M, Rahim F, Kavakli IH et al (2016) Milp-hyperbox classification for structure-based drug design in the discovery of small molecule inhibitors of SIRTUIN6. RAIRO Oper Res 50(2):387–400. https://doi.org/10.1051/ro/2015042
44. Tasci E, Zhuge Y, Kaur H et al (2022) Hierarchical voting-based feature selection and ensemble learning model scheme for glioma grading with clinical and molecular characteristics. Int J Mol Sci 23(22):14155. https://doi.org/10.3390/ijms232214155
45. Theodorakopoulos I, Kastaniotis D, Economou G et al (2014) Pose-based human action recognition via sparse representation in dissimilarity space. J Vis Commun Image Represent 25(1):12–23. https://doi.org/10.1016/j.jvcir.2013.03.008
46. Uddin MK, Lam A, Fukuda H et al (2021) Fusion in dissimilarity space for RGB-d person re-identification. Array 12:100089. https://doi.org/10.1016/j.array.2021.100089
47. Wang S, Liu Y, Xu Y, et al (2021) Want to reduce labeling cost? gpt-3 can help. https://doi.org/10.48550/ARXIV.2108.13487, https://arxiv.org/abs/2108.13487
48. Wanyan T, Lin M, Klang E, et al (2022) Supervised pretraining through contrastive categorical positive samplings to improve COVID-19 mortality prediction. In: Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. ACM, https://doi.org/10.1145/3535508.3545541
49. Yoon J, Zhang Y, Jordon J et al (2020) Vime: extending the success of self- and semi-supervised learning to tabular domain. In: Larochelle H, Ranzato M, Hadsell R et al (eds) Advances in neural information processing systems, vol 33. Curran Associates Inc., New York, pp 11033–11043
50. Zaidi A (2023) Predicting wildfires in algerian forests using machine learning models. Heliyon 9(7):e18064. https://doi.org/10.1016/j.heliyon.2023.e18064
51. Zhang X, Song Q, Wang G et al (2014) A dissimilarity-based imbalance data classification algorithm. Appl Intell 42(3):544–565. https://doi.org/10.1007/s10489-014-0610-5
52. Zottesso RH, Costa YM, Bertolini D et al (2018) Bird species identification using spectrogram and dissimilarity approach. Eco Inform 48:187–197. https://doi.org/10.1016/j.ecoinf.2018.08.007