Journal of Classification
https://doi.org/10.1007/s00357-024-09471-5

A Novel Classification Algorithm Based on the Synergy Between Dynamic Clustering with Adaptive Distances and K-Nearest Neighbors

Mohammed Sabri[1,2] · Rosanna Verde[2] · Antonio Balzanella[2] · Fabrizio Maturo[3] · Hamid Tairi[1] · Ali Yahyaouy[1] · Jamal Riffi[1]

Accepted: 18 April 2024

Abstract
This paper introduces a novel supervised classification method based on dynamic clustering (DC) and K-nearest neighbor (KNN) learning algorithms, denoted DC-KNN. The aim is to improve the accuracy of a classifier by using a DC method to discover the hidden patterns of the apriori groups of the training set. It provides a partitioning of each group into a predetermined number of subgroups. A new objective function is designed for the DC variant, based on a trade-off between the compactness and separation of all subgroups in the original groups. Moreover, the proposed DC method uses adaptive distances which assign a set of weights to the variables of each cluster, which depend on both their intra-cluster and inter-cluster structure. DC-KNN performs the minimization of a suitable objective function. Next, the KNN algorithm takes into account objects by assigning them to the label of subgroups. Furthermore, the classification step is performed according to two KNN competing algorithms. The proposed strategies have been evaluated using both synthetic data and widely used real datasets from public repositories. The achieved results have confirmed the effectiveness and robustness of the strategy in improving classification accuracy in comparison to alternative approaches.

Keywords K-nearest neighbors · Dynamic clustering · Combinatorial classification · Adaptive distances

1 Introduction

Classification is a fundamental task in machine learning, involving assigning data objects to apriori classes based on the values they assume for a set of features. It has received significant interest and has been extensively utilized in fields such as healthcare and medical diagnosis (Sivasankari et al., 2022; Malakouti, 2023), as well as image and video recognition (Wang et al., 2023; Chen et al., 2021).

The accuracy of a classifier depends on the availability of relevant and informative features, as well as on the choice of the classification algorithm. However, in many real-world scenarios, the data is complex, and the available features may not provide enough information to achieve high accuracy.

Extended author information available on the last page of the article

Published online: 11 May 2024

The statistical literature on supervised classification is growing rapidly. The use of a single algorithm often produces unsatisfactory results, often due to the complex structure of the groups to be classified (e.g., unbalanced distributions, non-linear relationships from predictors, presence of anomalous values). In recent years, techniques integrating or merging multiple algorithms from both supervised and unsupervised learning have been developed to enhance the decision rules provided by the model (Soheily-Khah et al., 2018; Sarker, 2021).

The K-nearest neighbor (KNN) algorithm has gained recognition as a powerful tool in the field of machine learning, providing an effective and straightforward method for classification in various pattern recognition scenarios (Zhang, 2016; Taunk et al., 2019). The primary approach employed by KNN involves determining the class of query samples by measuring the distance to the objects in the training set. The label of the query sample is then set by majority voting on the membership of the k-nearest objects in the training set. Recently, several novel adaptations of the KNN algorithm have been developed (Zhang et al., 2017b; Luo et al., 2020; Rastin et al., 2021a).

The Euclidean distance is often used with KNN algorithm to measure the dissimilarity between training and testing data. This procedure involves computing the dissimilarity, determining the nearest k neighbors based on these dissimilarities, and subsequently classifying the test sample based on the dominant class among the k neighbors. Although the Euclidean distance is easy to understand, it assigns equal importance to all sample features by considering them equally when calculating the distance. The use of equal weighting might be a limitation, particularly in situations where distinct features have differing degrees of importance to the categorization objective. To tackle this problem, various research has suggested alternative distance metrics that provide a more sophisticated method for calculating distances in KNN-based classification. These metrics have the potential to improve the performance of KNN-based classification (Chomboon et al., 2015; Ruan et al., 2021; Zhao & Yang, 2023).

This work explores the apriori classes and suggests the presence of subclasses or hidden patterns within them. This concept forms the core motivation of our research. The lack of exploration of such hidden patterns in existing literature strengthens our investigation. Although there have been numerous advancements in KNN recently (Gou et al., 2019a, b, c, 2022), none of them particularly address these hidden patterns.

The objective of our work is to bridge this gap by providing a comprehensive understanding of the complexities of apriori classes, with a specific emphasis on the unknown subclasses and hidden patterns that may exist within them.

In reality, many phenomena are often characterized by multiple sub-structures or sub-patterns. This implies that instances belonging to the same class can be distinguished by specific characteristics with varying relevance in the classification process.

The objective of this research is to improve the accuracy of the supervised classification (KNN) on high-dimensional data by integrating an unsupervised classification phase. The main idea is to extract relevant information from the original data by discovering sub-patterns that can aid in the classification task. However, integrating this information with a supervised classification algorithm in an efficient way poses a significant challenge.

This paper proposes a strategy based on three key points: (1) An algorithm based on the dynamic clustering (DC) algorithm (Diday, 1971) obtains subgroups from the initial labeled data which are combined with the original patterns to form a new cluster space. (2) An appropriate weight system is sought, aiming to find optimal weights for the features of each subgroup, using adaptive distances. (3) A KNN classification method that assigns the labels to data according to the cluster space carried out from the DC partition.

The traditional objective function used in the most well-known DC based method, the K-means, relies solely on the sum of the within-cluster deviance (Sinaga & Yang, 2020). This means that the objective function is related to the quadratic distances between each object in a cluster and the cluster representative. Despite awareness of Huygens' theorem, which states that minimizing the deviance within clusters is equivalent to maximizing the deviance between clusters, the optimized criterion is not designed to handle the constraints imposed by the presence of apriori groups, due to the labels in the training set.

To address these challenges, a new objective function is proposed for the DC method that relies on both inter-cluster and intra-cluster variability to improve the algorithm's performance and robustness. It is optimized to provide the identification of more homogeneous patterns (subgroups) and better separation between subgroups.

We also propose the integration of adaptive distances into the clustering procedure to measure the importance of features in the classification process, especially for complex, high-dimensional data (Diday et al., 1981). Weights are assigned to the features for each cluster. This results in the selection of features, according to the values of the associated weights, that significantly influence the achieved clusters (Li & Wei, 2020).

Finally, two supervised KNN classifier variants are proposed to label new elements according to the clusters of the achieved partitions of the initial classes. Specifically, the first proposal assigns a new instance to an apriori class based on the minimum (adaptive) distance to the elements of the clusters, while the second proposal considers the proximity to centroids (or representative elements) of the clusters rather than to the single elements, to improve significantly the computational cost, especially when dealing with a large number of elements.

The search for clusters (new patterns in a priori classes) through a DC algorithm improves the performance and accuracy of the classifier results.

This work has the ambition to advance our understanding of the dynamic clustering algorithm and provide an original approach in the field of classification. Especially, the proposal is denoted by two noteworthy innovations. Firstly, the paper highlights the benefits of using unsupervised classification techniques to identify new patterns in the original groups. This approach has a direct impact on the supervised classification's performance, and the combination of these techniques leads to more definitive predictive results. Secondly, the paper proposes an alternative objective function to be utilized during the clustering stage. The adoption of this approach is expected to yield richer results than minimizing only the between clustering.

The structure of this paper is as follows. Section 2 provides a review of the main contributions in the current literature. Section 3 presents the proposed classification approach and the combination between DC and KNN, as well as a new objective function to ensure homogeneity within subgroups of the original dataset. Furthermore, Sect. 4 provides an application of the suggested approach on real datasets, along with a simulation study that covers six distinct scenarios. Section 5 offers concluding remarks and discusses future directions for further investigation.

2 Literature Review

Classification techniques, now increasingly developed in the field of machine learning, address the problem of assigning entities to predefined classes. Most classification methods construct models based on features that represent the characteristics of prior classes.

Many of them have been proposed with the aim of selecting the most discriminating features of groups of individuals and carrying out stable classification rules to predict the behavior of new entities. The most traditional algorithms of classification as KNN (Fix & Hodges, 1989), Naive Bayes classifier (Duda et al., 2006), C4.5 (Quinlan et al., 1996), logistic regression, classification and regression tree (Breiman, 2017), and the stochastic gradient boosting decision tree (Friedman, 2002) are recognized as having high accuracy. Due to the increasing volume of data associated with many real-world problems and their inherent complexity, novel learning classifiers have been proposed. These include variants of KNN, such as k-most similar neighbor (k-MSN), linear scan, and locality-sensitive hashing (LSH). Other classification algorithms designed for high-dimensional data encompass extreme learning machine, sparse representation-based classification (Abavisani & Patel, 2019), and certainly, all deep learning algorithms.

Numerous studies have been carried out to compare different classification methods in order to select the most appropriate classifier for specific problems, among the various papers with this aim, see Zhang et al. (2017a). One of the main considerations in comparing classification methods is that performance depends on the data analyzed and not on the particular algorithm. Likewise, accuracy should not be understood as the only measure of algorithm performance. Strong attention must be paid to feature selection as it deals with dimensionality reduction.

One of the main challenges of classification methods regards feature selection. Classes are typically not distinguished by explicit features. Despite the use of advanced feature selection algorithms, the number of dimensions in these characteristics can still be very large, making it challenging to accurately capture the similarity of classes. The KNN technique, like other classical methods, has inherent limitations that restrict its classification capacity. The limitations cover the curse of dimensionality, the computational cost, sensitivity to outliers, the challenge of determining the optimal value for k, and its non-parametric nature, which affects the interpretation and generalization of results with new data. Although KNN is both simple and powerful, it is crucial to take into account these constraints when deciding if it is a suitable method for a certain task.

Numerous approaches have been proposed to overcome the limitations of traditional KNN by proposing new variants. To address the problem of the KNN algorithm's sensitivity to the choice of k, researchers have proposed different methods to dynamically determine the optimal k. One such approach, suggested by Gou et al. (2019b), involves two variations of the KNN rule: weighted representation-based KNN rule (WRKNN) and weighted local mean representation-based KNN rule (WLMRKNN). The experimental results indicate that the suggested methods demonstrate a lower sensitivity to the number of cluster k. The research conducted by Gou et al. (2019a) introduces the generalized mean distance-based KNN (GMD-KNN) classifier as a method to enhance the choice of the neighbor's number k. They asserted that the proposed technique shows lower sensitivity to the parameter k compared to the KNN-based classifiers. In Pan et al. (2020) propose a locally adaptive KNN algorithm based on discrimination class (AD-LAKNCN). This approach optimizes the values of k by taking into account the discrimination classes from the majority and second majority classes within the k neighborhood.

Furthermore, the KNN algorithm lacks a mechanism to assign varying weights to surrounding data points. To address this, Gou et al. (2012) introduced a new classification algorithm called distance-weighted KNN rule (DWKNN). This algorithm aims to overcome the sensitivity problem of selecting the neighborhood size and enhance classification performance. DWKNN utilizes a distance-weighted dual function and proves to be relatively robust to different choices of K, demonstrating good performance with a larger optimal K, as evidenced by experimental results on twelve real datasets. The performance of DWKNN surpasses other KNN-based methods currently considered state-of-the-art. Afterward, Rastin et al. (2021b) introduced a KNN stacking technique that employs a feature-weighted distance metric to mitigate the impact of irrelevant classes during stacking. Both of the aforementioned approaches take into account the weight of each adjacent point. However, it is important to note that when selecting weights, considering merely the distance information is inadequate.

In order to address the issue of the KNN algorithm's sensitivity to noise points, Gou et al. (2019c) introduced a KNN approach called LMRKNN, which utilizes the multi-local mean vectors of the KNN belonging to the same class to linearly represent the testing sample. The referenced method employs local mean vectors effectively to reduce outlier influence, achieving notable classification accuracy. However, its performance is still sensitive to the selection of the k parameter.

Cherif (2018) proposed a K-means-based-KNN algorithm that utilizes the K-means algorithm to partition the training dataset into a predetermined number of clusters. Subsequently, the centroids of each cluster are determined, resulting in a new training dataset consisting solely of these centroids. The 1-nearest neighbor algorithm is then applied to this new training dataset; therefore, the classification is achieved by selecting the closest neighbor in terms of distance. Uddin et al. (2022) provides a critical evaluation of different KNN variants, including the 1NN approach, in scenarios characterized by high levels of noise and outliers. The study's findings suggest that newer variants of KNN, potentially including those streamlined for efficiency like the K-means-based KNN, may not perform as effectively as the traditional KNN algorithm in complex, noisy environments. This underlines the importance of a careful selection of KNN variants depending on the specific characteristics of the dataset at hand, particularly when dealing with noise and outliers.

Maturo and Verde (2022) proposed a functional supervised classifier that combines functional data analysis with functional K-means and the functional KNN methods, improving the supervised classifier's accuracy in classifying ECG signals. In an application on medical data, the authors showed that a clustering of labeled data was able to detect false positives and false negatives in the classification of healthy and sick patients not so well identified by other classification methods even by ensemble ones.

It is important to note that while approaches that combine K-means and KNN exist, they do not specifically focus on the theoretical aspect to emphasize the distances between subgroups of partition inside the apriori classes. To take this into account, this research proposes a new objective function that maximizes the inter-cluster separation between the subclasses of an apriori cluster and all subclasses of the other apriori classes.

Inter-cluster separation is a crucial factor in ensuring that resulting clusters are meaningful and easily interpretable. The aim is to take into consideration the pattern structure of the data. This approach also revealed the presence of subsets of anomalous patterns within the classes, which are labeled in the same way even though they present different characteristics. Furthermore, various studies have investigated the application of weighting techniques to improve the performance of clustering algorithms.

As proposed by Diday and Govaert (1977), the concept of dynamic clustering with adaptive distances is to assign a distance to each cluster based on its intra-cluster structure. Recent developments in this area have focused on the use of adaptive distance metrics for symbolic data, such as multivariate aggregated data in the form of intervals, histograms, and other data types. Incorporating an adaptive distance metric in clustering algorithms can improve their performance in various aspects. For example, outliers or anomalous data can have a significant impact on the determination of the centroids, but by assigning lower weights to such points, their influence can be reduced, making the clustering process more robust to outliers.

Adaptive distances for classical data have been mainly defined as Euclidean-weighted distances. Recent advancements in symbolic data analysis have led to the development of a range of adaptive distance metrics customized for DC on aggregated data. Notably, De Carvalho and Lechevallier (2009) introduced the adaptive City-Block and Hausdorff distances for the partition-based clustering of symbolic interval data. Furthermore, the squared Wasserstein distance, which is specifically designed for histogram data, has been described in detail in Irpino et al. (2014); Balzanella and Verde (2020). Moreover, Rodríguez and de Carvalho (2022) have contributed to this field by developing adaptive Euclidean and City-Block distances for interval-valued data.

Bao et al. (2018) presented a new approach for addressing interval-valued data clustering, wherein they proposed an adaptive fuzzy c-means algorithm that incorporates the consideration of interval membership across various clusters within the partition. de Carvalho et al. (2022) presented a batch self-organizing map (SOM) algorithm for distributional-valued data based on a weighted Wasserstein distance, where the weights are computed through the optimization of the clustering loss function.

3 Methodology

This section presents two novel approaches to supervised classification using clustering. The first step involves developing a new DC variant to cluster apriori classes. The approach incorporates a new objective function that employs intra-cluster compactness, which uses an adaptive distance metric to compute dispersion information between subgroups of a given class, as well as inter-cluster separation, which measures the distance between a subgroup of a specific class and the subgroups of other apriori classes. Finally, the results obtained from DC are used for classification with the new KNN algorithm. In this process, subgroup weights are utilized with the adaptive distance to measure the similarity between the testing sample and their neighboring subgroup centroid points. Subsequently, the k nearest neighbors are determined based on the calculated similarities. The allocation of the new instances to a class is based on the majority vote among the neighbors of the k subgroup centroids.

3.1 Dynamic Clustering Algorithm

Clustering is a widely utilized technique in various applications, including image processing (Chang et al., 2017), video processing (Alayrac et al., 2016), gene analysis (Dapas et al., 2020), healthcare (Liao et al., 2016), and community detection (Li et al., 2022), among others. It involves dividing a dataset into groups, or clusters, based on similarity criteria, where objects within the same cluster are more alike than those in different clusters.

In this paper, our focus is on DC, an unsupervised learning algorithm that aims to partition data into clusters while simultaneously finding cluster representatives consistent with the distance function used for allocating units. Typically, the representatives are obtained as the minimizers of the sum of distances. The classic K-means algorithm can be seen as a specific case of DC where the distance metric is the Euclidean distance, and the representatives (centroids) are calculated as cluster averages.

The original concept of dynamic clustering was introduced by Diday (1971) and involves a two-step process of constructing clusters and selecting the best prototype for each cluster based on an adequacy criterion (Diday & Simon, 1976). The advantages of this scheme are mainly its flexibility with respect to the nature of the analyzed data and the choice of the distance function and the focus on providing cluster representatives, named prototypes. For instance, DC methods exist for datasets described by interval variables (De Carvalho & Lechevallier, 2009) and histogram variables (Balzanella & Verde, 2020; de Carvalho et al., 2022).

Let $X = \{X_1, \ldots, X_i, \ldots, X_n\}$ be a set of n objects, where each object $X_i = \{x_{i1}, \ldots, x_{im}\}$ is described by a set of m features. The general DC looks for the partition $G = \{C_1, \ldots, C_K\}$ in K clusters and the set $Z = \{Z_1, \ldots, Z_K\}$ of K prototypes representing the clusters in G, such that the following Δ fitting criterion between the set Z of prototypes and the partition G is minimized:

$$\Delta(G, Z) = \sum_{k=1}^{K} \sum_{X_i \in C_k} d(X_i, Z_k) \tag{1}$$

The fitting criterion is defined as the sum of dissimilarities or distance measures between each object X_i belonging to a class $C_k \in G$ and the class representation $Z_k \in Z$.

In this context, the DC algorithm iteratively implements the following representation and allocation steps:

1. The representation step describes the K clusters $(C_1, \ldots, C_K)$ of the partition G through a vector $Z = (Z_1, \ldots, Z_K)$ of prototypes. Keeping the partition $G = \hat{G}$ fixed for the current iteration of the algorithm, Z is obtained from the minimization of $\Delta(\hat{G}, Z)$, which is equivalent to finding the $Z_k \ \ (k = 1, \ldots, K)$ that minimize $\sum_{i \in C_k} d(X_i, Z_k)$.
2. The allocation step assigns each element X_i to a cluster C_k according to the proximity to the prototype $Z_k \in Z$. Keeping $Z = \hat{Z}$ fixed for the current iteration of the algorithm, it finds the partition of G that minimizes $\Delta(G, \hat{Z})$, by finding the cluster $C_k = \{X_i \in X \mid d(X_i, Z_k) \leq d(X_i, Z_l), \forall l = 1, \ldots, K; l \neq k\}$.

3.1.1 DC as a Generalization of K-Means Algorithm

The K-means clustering methodology is broadly utilized as a partitioning strategy. The proposed dynamic clustering method is a generalization of the K-means algorithm, which puts forth a compelling notion that cluster centers do not essentially have to be the centroids of clusters in R^m. Rather, it suggests substituting them with centers that can take various forms, based on the problem that needs to be addressed.

The K-means algorithm starts by selecting K initial cluster centers and then assigns each object to the closest cluster through the optimization of an objective function. As mentioned previously, the classical K-means algorithm only considers the intracluster compactness and the distances between the cluster centroids and individual data points. The membership matrix U, a $n \times K$ binary matrix, indicates which objects are assigned to which clusters, and $Z = \{Z_1, \ldots, Z_k, \ldots, Z_K\}$ represents the centroids of the K clusters, with elements $Z_k = \{z_{k1}, \ldots, z_{kj}, \ldots, z_{km}\}$ for each feature $j = 1, \ldots, m$.

The objective function of the classic K-means without considering the inter-cluster separation is the following:

$$\Delta(U, Z) = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ip} \sum_{j=1}^{m} (x_{ij} - z_{kj})^2, \tag{2}$$

such that $\sum_{k=1}^{K} u_{ik} = 1$, (with $u_{ik} \in \{0, 1\}$), and $X_i = \{x_{i1}, \ldots, x_{ij}, \ldots, x_{im}\}$ is an object of X described by m features.

DC algorithm optimizes the objective function by alternating the *representation* and *allocation* steps:

1. *Representation step* (the matrix of membership $\hat{U}$ is fixed)
   The solution for the optimization problem $\Delta(\hat{U}, Z)$ is provided by the minimizer Z
   $$z_{kj} = \frac{\sum_{i=1}^{n} u_{ik} x_{ij}}{\sum_{i=1}^{n} u_{ik}}, \tag{3}$$
   where $1 \leq k \leq K$
2. *Allocation step* (the vector of centroids $\hat{Z}$ is fixed)
   According to Chan et al. (2004), the minimizer U of the optimization problem $\Delta(U, \hat{Z})$ is given by
   $$u_{ik} = \begin{cases} 1 \text{ if} \sum_{j=1}^{m} (x_{ij} - z_{kj})^2 \leq \sum_{j=1}^{m} (x_{ij} - z_{lj})^2, \\ 0 \text{ otherwise.} \end{cases} \tag{4}$$

The partitioning criterion (2) decreases at each iteration, converging to a stationary value.

3.1.2 The Need for Adaptive Distances in DC

The central concept of dynamic clustering with adaptive distances is to assign a specific distance measure, denoted as d_k, to each cluster C_k and to minimize the sum of distances $d_k(X_i, Z_k)$ between objects X_i belonging to cluster C_k and the centroid Z_k. Importantly, the distances employed in the DC algorithm are not fixed in advance but rather are tailored to each cluster.

In this clustering algorithm, a weighting step is introduced. It assigns a weight to each variable for each cluster, reflecting the relevance of the variable in a cluster. The use of adaptive distance can also be viewed as a means of automatically scaling variables, as scaling can greatly impact the dissimilarity values and clustering outcomes in clustering analysis.

The DC criterion, which incorporates adaptive distances, is expressed as follows:

$$\Delta(U, W, Z) = \sum_{k=1}^{K} \sum_{X_i \in C_k} u_{ik} d_k(X_i, Z_k), \tag{5}$$

such that $u_{ik} \in \{0, 1\}$, $\sum_{k=1}^{K} u_{ik} = 1$

In this context, distance d_k is a weighted sum of distances $d_{w_{kj}}$

$$d_k(X_i, Z_k) = \sum_{j=1}^{m} d_{w_{kj}}(x_{ij}, z_{kj}) = \sum_{j=1}^{m} w_{kj} d(x_{ij}, z_{kj}) \tag{6}$$

The adaptivity of the distance $d_{w_{kj}}$ is expressed by the vector of weights W_k.

When using adaptive distances, the representation step is divided in two stages so that the global optimization scheme is

1. Representation step
   (a) Stage 1: fix the matrix $\hat{U}$ of membership and the vector of weights $\hat{W}$
   Find the solution $Z_k = \{z_{k1}, \ldots, z_{km}\}$ of the optimization problem $\Delta(\hat{U}, \hat{W}, Z)$.
   (b) Stage 2: fix the matrix $\hat{U}$ of membership and the vector of centroids $\hat{Z}$
   Find the vector of weights $W_k = \{w_{k1}, \ldots, w_{km}\}$ that minimizes the criterion $\Delta(\hat{U}, W, \hat{Z})$.
2. Allocation step
   Fix the set of vectors of weights $\hat{W}$ and the set of vectors of centroids $\hat{Z}$. Find the membership matrix U that minimizes the criterion $\Delta(U, \hat{W}, \hat{Z})$

The paper employs an adaptive distance metric, specifically a weighted Euclidean distance, to calculate the distance between subgroups within a given cluster. Explicit formulas for the optimum cluster centroids, as well as for the weights of the adaptive distances, are found based on a new objective function criterion. By integrating the procedures of data partitioning and centroids selection with adaptive distances, DC algorithm provides a comprehensive and flexible approach to clustering analysis for apriori classes.

3.2 Dynamic Clustering Algorithm to Partition Apriori Groups

In this section, a new objective function is proposed to discover new information on the original data by combining both intra-cluster compactness of the subgroups of the same apriori group and inter-cluster separation between one subgroup and the subgroups of other apriori classes, as illustrated in Fig. 1. Therefore, it may be ineffectual to evaluate the weights

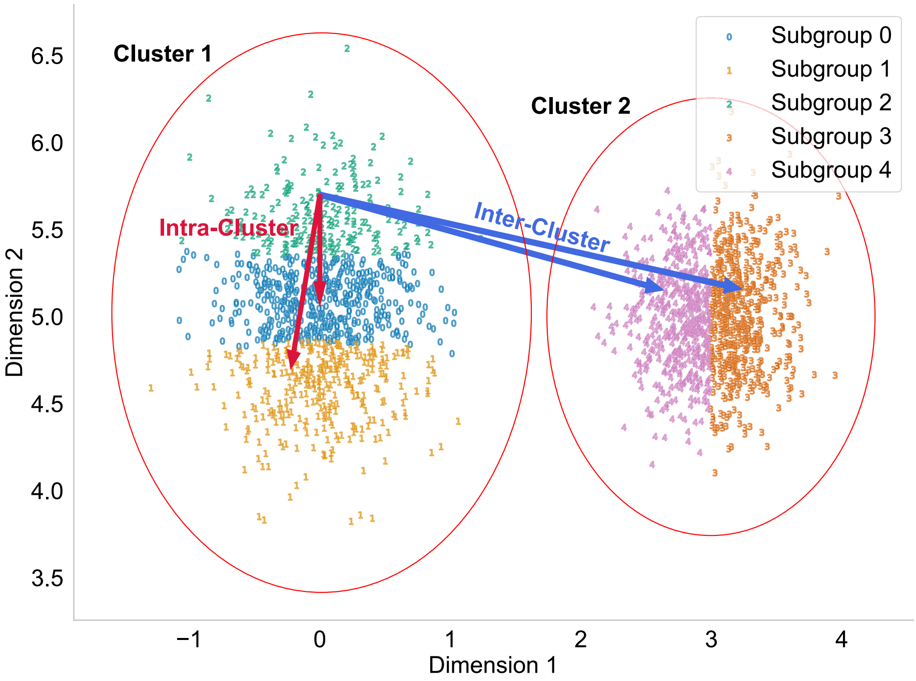


Fig. 1 Scatter plot illustrating subgroups within two apriori groups

of the variables of a subgroup using only the variation within the groups of a data set. Under these conditions, inter-cluster separation can play a significant role in differentiating the significance of various patterns and taking into account the heterogeneity among the subgroups of each original group.

We apply inter-cluster separation by introducing the global subgroups centroids of a data set. In contrast to the conventional DC, our proposed DC algorithm maximizes the distances between the subgroup's centroid of an apriori group and the global subgroups centroid of the other apriori groups partition, while minimizing the distances between objects and their subgroups centroid.

Let N be the total number of apriori classes, and $U = \{U_1, \ldots, U_g, \ldots, U_N\}$ the set of N matrices. Let n_g be the number of elements of class g and c_g be the number of subgroups in the class g. Each U_g is an $n_g \times c_g$ indicator matrix containing the membership of each element i of the apriori class g to the subgroup p; such that where $u_{gip} = 1$ denotes that the i-th object belonging to group g is assigned to subgroup p; otherwise, $u_{gip} = 0$, indicating that the object is not assigned to subgroup p. Let $Z = \{Z_1, \ldots, Z_g, \ldots, Z_N\}$ be a set of vectors representing the centroids of each original group. For group g, let $Z_g = \{Z_{g1}, \ldots, Z_{gc_g}\}$ be a set of c_g vectors that represent the subgroups' centroids and let $W_g = \{W_{g1}, W_{g2}, \ldots, W_{gc_g}\}$ be a set of weight vectors associated with the subgroups, where w_{gpj} represents the weight of the j-th variable related to the p-th subgroup for class g. Let β represent a parameter used for adjusting the weights.

With the aim of achieving both intra-cluster compactness and inter-cluster separation, the optimization process is performed using a DC algorithm in which the objective function is modified to emphasize the separation between clusters belonging to different apriori classes:

$$\begin{aligned} P(U, W, Z) &= \sum_{g=1}^{N} \sum_{p=1}^{c_g} \frac{\sum_{i=1}^{n_g} u_{gip} d_g^2(X_i, Z_{gp})}{n_g d^2(Z_{gp}, Z_{gG})} \\ &= \sum_{j=1}^{m} (\sum_{g=1}^{N} \sum_{p=1}^{c_g} \frac{w_{gpj}^{\beta} \sum_{i=1}^{n_g} u_{gip}(x_{ij} - z_{gpj})^2}{n_g (z_{gpj} - z_{gGj})^2}), \end{aligned} \tag{7}$$

such that $u_{gip} \in \{0, 1\}$, $\sum_{p=1}^{c_k} u_{gip} = 1$, and $\sum_{j=1}^{m} w_{gpj} = 1$.

In the context of our study, the distance metric d_g is defined as a weighted sum of distances $d_{w_{gpj}}$, where $d_{w_{gpj}}$ represents the distance metric for the p-th subgroup of the g-th apriori class. The vector of weights W_{gp} demonstrates the adaptivity of the distance metric $d_{w_{gpj}}$:

$$d_g(X_i, Z_{gp}) = \sum_{j=1}^{m} d_{w_{gpj}}(x_{ij}, z_{gpj}) = \sum_{j=1}^{m} w_{gpj}(x_{ij}, z_{gpj})^2. \tag{8}$$

Let us assume that the present group is the g^{th} group. z_{gGj} represents the j^{th} feature of the global subgroups centroid of all other apriori groups, excluding the current group g.

We calculate z_{gGj} as

$$z_{gGj} = \frac{\sum_{h \in \{1,\ldots,N\} \setminus \{g\}} c_h \sum_{q=1}^{c_h} z_{hqj}}{c_1 + \cdots + c_{g-1} + c_{g+1} + \cdots + c_N}. \tag{9}$$

To initiate the solution process of the objective function, it is necessary to initialize the parameters $\hat{U}$, $\hat{W}$, and $\hat{Z}$ of all groups (for $g = 1, \ldots, N$). Subsequently, the partition of the group g is evaluated, thus reducing the minimization issue as

$$P(U, W, Z) = \sum_{p=1}^{c_g} \sum_{i=1}^{n_g} u_{gip} \sum_{j=1}^{m} w_{gpj}^{\beta} \frac{(x_{ij} - z_{gpj})^2}{n_g (z_{gpj} - z_{gGj})^2}, \tag{10}$$

such that $u_{ip} \in \{0, 1\}$, $\sum_{p=1}^{c_g} u_{gip} = 1$, and $\sum_{j=1}^{m} w_{gpj} = 1, 1 \leq p \leq n_g$.

To minimize (10), it is necessary to solve the problems $P1$, $P2$, and $P3$ iteratively.

1. Specifically, the representation step requires solving two distinct problems $P1$ and $P2$:
   Problem $P1$: fix $U = \hat{U}$, $W = \hat{W}$ and solve the reduced problem $P(\hat{U}, Z, \hat{W})$
   Problem $P2$: fix $U = \hat{U}$, $Z = \hat{Z}$ and solve the reduced problem $P(\hat{U}, \hat{Z}, W)$
2. To address the allocation problem, it is necessary to solve the problem denoted as $P3$:
   Problem $P3$: fix $Z = \hat{Z}$, $W = \hat{W}$ and solve the reduced problem $P(U, \hat{Z}, \hat{W})$

To solve the problem $P1$, we calculate the gradient of P with respect to z_{gpj} as

$$\frac{\partial P(\hat{U}, \hat{W}, Z)}{\partial z_{gpj}} = -2 w_{gpj}^{\beta} \sum_{i=1}^{n_g} u_{gip} \frac{(x_{ij} - z_{gpj})(z_{gpj} - z_{gGj})^2 + (z_{gpj} - z_{gGj})(x_{ij} - z_{gpj})^2}{n_k (z_{gpj} - z_{gGj})^4}; \tag{11}$$

by setting (11) to zero, we have:

$$z_{gpj} = \frac{\sum_{i=1}^{n_k} u_{gip} x_{ij} (x_{ij} - z_{gGj})}{\sum_{i=1}^{n_g} u_{gip} (x_{ij} - z_{gGj})}. \tag{12}$$

The initial section of the supplementary material contains the proof and the necessary and sufficient conditions required for the realization of this finding.

It is worth noticing that z_{gpj}, the representative (e.g., centroid) of C_{gp}, can be interpreted as a weighted average of the elements of the p^{th} subgroup, with weights being the difference between x_{ij} and the global centroid z_{gG_j} computed as in Eq. 9 on all other apriori groups, excluding the current group.

The higher the difference, the more the subgroup element contributes to the determination of the subgroup's centroid. This result is due to the optimization of the discriminant component of the criterion which emphasizes the separation between classes.

The internality condition of the centroid z_{gpj} of the subgroup C_{gp} (for each variable j) is guaranteed under the conditions demonstrated in the Appendix, whereas it can become external to the cluster interval of values (for each j) the closer z_{gGj} is to the mean of the elements of the cluster C_{gp}.

The problem $P2$ will be solved by setting up a Lagrangian equation to $P(\hat{U}, \hat{Z}, W)$ with multiplier λ. Let $L(W, \lambda)$ be the Lagrangian

$$L(W, \lambda) = \sum_{p=1}^{c_g} \sum_{j=1}^{m} w_{gpj}^{\beta} D_{gpj} - \lambda (\sum_{j=1}^{m} w_{gpj} - 1), \tag{13}$$

where $D_{gpj} = \sum_{i=1}^{n_k} u_{gip} \frac{(x_{ij}-z_{gpj})^2}{n_g(z_{gpj}-z_{gGj})^2}$. Setting the gradient of Eq. 13 with respect to w_{gpj} and λ to zero, we obtain

$$\frac{\partial L(W,\lambda)}{\partial w_{gpj}} = \beta w_{gpj}^{\beta-1} D_{gpj} - \lambda = 0; \tag{14}$$

from Eq. 14, we obtain

$$w_{gpj} = \left(\frac{\lambda}{\beta D_{gpj}}\right)^{\frac{1}{\beta-1}}. \tag{15}$$

The gradient with respect to λ

$$\frac{\partial L(W,\lambda)}{\partial \lambda} = -(\sum_{j=1}^{m} w_{gpj} - 1) = 0; \tag{16}$$

substituting (15) into (16), we obtain

$$\lambda^{\frac{1}{\beta-1}} = \frac{\beta^{\frac{1}{\beta-1}}}{\sum_{j=1}^{m} D_{gpj}^{-\frac{1}{\beta-1}}}; \tag{17}$$

substituting (17) into (15), we have

$$w_{gpj} = \frac{1}{(D_{gpj})^{\frac{1}{\beta-1}} \sum_{l=1}^{m} D_{gpl}^{-\frac{1}{\beta-1}}}, \tag{18}$$

The minimizer W_k of the optimization problem $P2$ is given by

$$w_{gpj} = \begin{cases} 0 & \text{if } (z_{gpj} - z_{gGj})^2 = 0, \\ 0 & \text{if } D_{gpj} \neq 0, \quad \text{but} \quad D_{gpl} = 0, \quad \text{for some } l, \\ \dfrac{1}{(D_{gpj})^{\frac{1}{\beta-1}} \sum_{l=1}^{m} D_{gpl}^{-\frac{1}{\beta-1}}} & \text{otherwise.} \end{cases} \tag{19}$$

The problem $P3$ is solved by

$$u_{gip} = \begin{cases} 1 \text{ if} \sum_{j=1}^{m} w_{gpj}^{\beta} \dfrac{(x_{ij} - z_{gpj})^2}{n_g(z_{gpj} - z_{gGj})^2} \leq \sum_{j=1}^{m} w_{gpj}^{\beta} \dfrac{(x_{ij} - z_{grj})^2}{n_g(z_{grj} - z_{gGj})^2}, \\ 0 \text{ otherwise,} \end{cases} \tag{20}$$

where $1 \leq r \leq g, r \neq p$.

The same process is provided to the partition of the other groups $q \neq g$ for $q = 1, \ldots, N$ of the primary datasets, by optimally computing u_{lip}, z_{lpj}, and w_{lpj}.

3.3 New DC Variant Algorithm

In this section, we provide a comprehensive explanation of the algorithm used in the novel DC variant clustering method. The aim of this algorithm is to create new subgroups from the available labeled data by detecting hidden patterns.

The algorithm is designed to associate a unique distance metric with each cluster, which is used to compare clusters and their representatives. This distance measure is not fixed and varies from subgroup to subgroup, changing with each iteration until convergence. The adaptive nature of this distance measure offers the advantage of assigning weights to the variables that are more representative or informative of a particular cluster, resulting in a more accurate clustering algorithm.

This adaptive approach aims to identify a partition of each original class, denoted as $G_1, \ldots, G_N$, respectively into $n_{c_1}, \ldots, n_{c_N}$ subgroups. Here, $G_g = \{C_{g1}, \ldots, C_{gn_{c_g}}\}$ specifies the partitions for group g, and the corresponding centroids, denoted as $Z_g = \{Z_{g1}, \ldots, Z_{gn_{c_g}}\}$, for each group are computed using the formula for centroids (12). Additionally, for each subgroup, a set of weights is assigned from the set $W_g = \{W_{g1}, \ldots, W_{gn_{c_g}}\}$.

The algorithm we propose looks for a local minimum of the objective function in Eq. 7.

It requires, as an input, the training dataset, with N apriori classes, and the number of subgroups for each apriori class. It starts from an initial random partitioning of the apriori classes into subgroups, then initializes the weights of variables for each subgroup to $1/m$ (where m is the number of variables). An initial set of centroids Z is computed according to Eq. 12, based on the initial random partition and on the weights in W.

The iterative part of the algorithm alternates, at each iteration t, the representation and allocation step introduced in Sect. 3.2, in order to provide the partition of the N apriori classes $G_1, \ldots, G_N$, into $n_{c_1}, \ldots, n_{c_N}$ subgroups; the set of centroids Z; the weights W.

At each iteration t, a check of the convergence of the algorithm is performed by evaluating the criterion P_t:

$$P_t = \sum_{g=1}^{N} \sum_{p=1}^{c_g} \frac{\sum_{i=1}^{n_g} u_{gip} d_g^2(X_i, Z_{gp})}{n_g d^2(Z_{gp}, Z_{gG})}, \tag{21}$$

where X_i, Z_{gp}, and u_{gip} are defined as before.

The algorithm will run when $\| P_{t+1} - P_t \| > 0$, which means that a further iteration improves the criterion. In other words, the algorithm continues as long as there is a decrease in the intra-cluster distances between subgroups of an original group and/or an increment in the inter-cluster distances.

Algorithm 1 displays the pseudocode of the suggested DC clustering algorithm.

3.4 KNN Classifier Based on Adaptive Distances and Novel Patterns

The K-nearest neighbor is a supervised learning technique that uses training data and a predetermined k value to find the k nearest data based on the idea of using distance computation to discover the nearby points of the query from the training set and assign a class label to the query through the majority voting rule. However, its efficacy is comparable to the most complex classifiers in the literature. This classifier relies heavily on measuring the distance or similarity between the tested examples and the training examples. This raises an essential question about which distance or similarity measures should be used for the KNN classifier out of the numerous options available. Therefore, we propose an adaptive distance parameterized by weight vectors. The weights are estimated during the first clustering step on apriori classes so that each subgroup is associated with its weight vector. The main idea of the KNN classifier with adaptive distances is that there is a distance to compare the test objects and their nearest points from the training dataset, which changes with each training point. However, the objects belonging to the same subgroup have the same vector weights.

Algorithm 1 Weighted dynamic clustering algorithm with adaptive euclidean distance.

```
1: **Input :**
2: $X = \{X_1, X_2, \ldots, X_n\}$: training dataset with N class
3: $c_1, \ldots, c_N$: optimal number of subclasses within each initial class
4: **Output :** $G = \{G_1, \ldots, G_N\}$ wherein each element $G_g$ constitutes a set of subclasses consisting of $c_g$
   elements, denoted as $G_g = \{C_{g1}, \ldots, C_{gc_g}\}$
5: Assign $x_i \in G_j (j = 1, \ldots, N)$ with random cluster labels;
6: Initialize $Z$ and $W$;
7: Calculate $P_t$ according to Eq. 21;
8: **while** $\|P_{t+1} - P_t\| > 0$ **do**
9:     **for** each $x_i \in G_g$ **do**
10:        Move $x_i$ from the current cluster to $C_l$ such that
               $l = \underset{k}{\operatorname{argmin}} d_w(x_i, Z_{gk}) = \underset{k}{\operatorname{argmin}} \sum_{j=1}^{m} w_{gkj}(x_{ij} - z_{gkj})^2.$
11:        Update $Z_{gl}$ using the Eq. 12;
12:        Update $W_{gl}$ using the Eq. 19;
13:     **end for**
14:     Calculate $P_{t+1}$ according to Eq. 21;
15:     Repeat the steps 9 to 13 to recalculate $G_f$, $Z_f$, and $W_f$ for $f \neq g$ $(g = 1, \ldots, N)$
16:     $t \longleftarrow t + 1$
17: **end while**
```

Let $T = (X_i, y'_{ip})_{i=1}^{n}$ represent a training set consisting of n training instances, with each instance belonging to one of N classes. Each training instance X_i is an element of a m-dimensional space R^m, and its corresponding class label y'_{ip} is obtained from the initial clustering step, with p represents the subgroup of X_i in the original apriori class y_i. When a new query S_h is given, we first compute the adaptive distances between S_h and each training instance in T. The adaptive distance for a given data point X_i and query S_h is defined as follows:

$$d_{y'_{ip}}(X_i, S_h) = d_{W_{y_i p}}(X_i, S_h) = \sum_{j=1}^{m} w_{y_i pj}(x_{ij} - s_{hj})^2. \tag{22}$$

Here, $W_{y_i p}$ is a vector of weights corresponding to the p-th subgroup of the apriori class y_i.

The n distances are then arranged ascendingly. $N_K(S_h) = \{(X_j, y'_{jp})\}_{j=1}^{K}$ denotes the K-nearest neighbors of S_h, which are the K training instances with the top K smallest distances. Ultimately, the majority voting rule is used to assign the query S_h to subgroup C_{gp}:

$$C_{gp} = \underset{C_{ip'}}{\operatorname{argmax}} \sum_{(x,y) \in N_k(S_h)} \mathbb{1}_{C_{ip'}}(y), \quad i = 1, \ldots, N \;\; and \;\; p' = 1, \ldots, c_i \tag{23}$$

where $\mathbb{1}_C(.)$ is the indicator function:

$$\mathbb{1}_C(y) = \begin{cases} 1 & \text{if } y \in C, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

3.5 The DC-KNN Combined Algorithm

DC-KNN (dynamic clustering and K-nearest neighbors) is an algorithm that combines the efficiency of the DC algorithm with the classification by KNN. The basic idea behind DC-KNN is to use DC with adaptive distances to re-cluster the classes of the training dataset into different subgroups and then use KNN to classify the test set based on the new labels obtained from the clustering step.

The algorithm starts by using DC to cluster each original group into a specific number of clusters; the optimal number of subgroups can be determined by the Silhouette or Elbow method. The resulting clusters will be used as preprocessing for the KNN algorithm, allowing it to work on more homogeneous subsets of data. This combination improves the accuracy and efficiency of the KNN algorithm, as it enables the algorithm to learn all the patterns of the dataset necessary for its learning process.

After the clustering step, KNN is used to classify new data points based on the newly discovered patterns. The KNN algorithm finds the K nearest neighbors of a given data point and determines the class of the majority of those neighbors. In this way, the DC-KNN algorithm is able to effectively combine the strengths of both DC and KNN, making it a powerful tool for data classification.

It is worth noting that the DC-KNN algorithm can be sensitive to the initial partitions. Hence, choosing the appropriate number of clusters and setting the parameters for the DC algorithm is essential.

The execution of the DC-KNN classification algorithm follows the steps outlined in Algorithm 2.

Algorithm 2 DC-KNN combining algorithm.

```
Input:
$X = \{X_1, \dots, X_i, \dots, X_n\}$: training dataset with $N$ class
$c_1, \dots, c_N$: optimal number of subclasses within each initial class
$k$: number of nearest number for the KNN algorithm
$S_l$: new instance
Output: $y_{S_l}$ the label of $S_l$
Step 1- Dynamic clustering algorithm: partition of the $N$ classes of $X$ respectively into $c_1, \dots, c_N$ subgroup using Algorithm 1
return: $W, Y'$
Step 2- KNN classifier
 Calculate the distance between $S_l$ and $X_i$ $(i = 1, \dots, n)$ using Eq. 22
 Select the subset $N_k$ from the dataset $X$, which contains the $k$ nearest neighbors of the sample $S_l$
     $N_k(S_l) = \{(X_i, y'_{ip})\}_{i=1}^{k}.$     (25)
 Calculate the subgroup $C_{gp}$ of $S_l$ according to Eq. 23
 Assign the label of the new element to its original class
     $y_g \leftarrow y'_{gp}$
```

3.6 The DC-KNN Using the Centroids as the Nearest Neighbors

A new variant of the KNN algorithm is introduced in this study for more accurate predictions of new data points based on the new subgroups. The proposed classification approach focuses on classifying instances to their nearest neighbor class by computing the distance between new instances and subgroups centroids. The allocation is then determined using the label of the apriori class to which the subgroups belong.

The DC-KNN classifier algorithm utilizes the centroids and weights of DC to determine the nearest neighbor of a new element. Here, K denotes the number of neighborhoods of centroids that are closest to the new query. The optimum value of K is chosen from the range of 1 to $(\min_{1 \leq i \leq N}(c_i) + 1)$. The algorithm follows the sequential steps outlined in Algorithm 3.

Algorithm 3 DC-KNN combining algorithm using centroids as NN.

```
Input:
$X = \{X_1, \dots, X_i, \dots, X_n\}$: training dataset with $N$ class
$c_1, \dots, c_N$: optimal number of subclasses within each initial class
$K$: number of nearest number for the KNN algorithm
$S_l$: new instance
Output : $G_l$ the original class in which $S_l$ belongs
Step 1- DC algorithm: partition of the $N$ classes of $X$ into $c_1, \dots, c_N$ using Algorithm 1
return: $Z, W$
Step 2- KNN classifier using the centroids as NN
Calculate the distance between $S_l$ and $Z_{gp}$ $(g = 1, \dots, N \ and \ p = 1, \dots, c_g)$
    $d^2(S_l, Z_{gp}) = \sum_{j=1}^{m} w_{gpj}(s_{ij} - Z_{gpj})^2$
Select the subset $N_k$, which contains the $K$ nearest neighbors centroid of the new sample $S_l$
    $N_K(S_l) = \{Z_i \ / \ Z_i \in Z_g, \ \ g = 1, \dots, N\}_{i=1}^{K}$
Calculate the belonging to the original class $G_l$ for $S_l$ such that
    $l = \underset{g}{\text{argmax}} \sum_{Z_{g'p'} \in N_K(S_l)} \mathbb{1}_{C_{gp}}(Z_{g'p'}), \quad for \ all \ g = 1, \dots, N \ \ and \ \ p' = 1, \dots, c_g \qquad (26)$
```

4 Experiments

This section conducts extensive experiments on different real and synthetic datasets to validate the classification performance of the proposed DC-KNN classifiers. The DC-KNN approach is compared to the KNN and Kmeans-KNN methods, in terms of classification accuracy.

4.1 Experimental Results on Real Datasets

To thoroughly assess the performance and robustness of the proposed DC-KNN algorithms, experiments are conducted comparing them with classical KNN and Kmeans-KNN. The latter uses the K-means clustering algorithm as a first step and then applies KNN using the results of the clustering. The comprehensive experiments are conducted on real datasets sourced from the UCI Machine Learning Repository Bache and Lichman (2013), KEEL attribute noise datasets Alcala-Fdez et al. (2011), and UCR Time Series Classification Repository Dau et al. (2018). The classification accuracy was used to measure the performance of all the approaches in each experiment. Note that DC-KNN1 represents Algorithm 2 that employs data points as nearest neighbors. It is important to note that in the KNN algorithm, the selection of the k nearest neighbors involves an aggregation from all available classes of data points. Similarly, in the KNN algorithm with classical K-means and DC-KNN1 algorithms, the nearest neighbors are selected from the data points of all the subgroups of the apriori classes. However, in the DC-KNN2 algorithm, the nearest neighbors are the centroids of the subgroups obtained from DC.

The objective of the experiments is to demonstrate the powerful classification capabilities of the proposed techniques on different kinds of datasets that represent real datasets, noisy numerical datasets obtained from the KEEL machine learning repository, and time series datasets from the UCR database. The utilized datasets mentioned here are clearly outlined

Table 1 Experimental datasets from UCR, UCI, and KEEL repositories used in the study

datasets	Total samples	Features	Classes	Testing samples
Breast	569	30	2	171
ILPD	583	10	2	175
Computers	250	720	2	250
ScreenType	375	720	3	375
StarLightCurves	1000	1024	3	8236
ItalyPowerDemand	67	24	2	1029
Yeast-n	1212	8	2	606
Sonar-n	208	60	2	66
Iono-n	351	34	2	71
Heart-n	270	13	2	54
Pima-n	768	8	2	154
Spambase-n	4597	57	2	920
Iris-n	150	4	3	60

in Table 1. Their information shows variations in the quantities of total samples, features, classes, and test samples.

We utilize the Wisconsin Breast Cancer Wisconsin (Diagnostic) dataset, abbreviated as “Breast,” “ILPD,” and the noise “Yeast” datasets from the UCI database. Additionally, we perform tests on the“’Computers,” “ScreenType,” and “StarLightCurves” datasets from the UCR repository for our time series analysis. Following the approach employed in Maturo and Verde (2022), we utilize functional data analysis to describe the time series datasets and extract the coefficients of the b-spline decomposition as features. The six noise datasets from the KEEL repository are “Sonar,” “Iono,” “Heart,” “Pima,” “Spambase,” and “Iris.” The experiments employ the abbreviations Yeast-n, Sonar-n, Iono-n, Heart-n, Pima-n, Spambase-n, and Iris-n to distinguish the noisy data. The six noise datasets consist of samples that are exposed to a noise intensity of 10%. To clarify, around 10% of the samples in each dataset are chosen at random, and around one-third of the total samples from each dataset are chosen as test samples, while the rest of the samples are designated as training samples. The values of a particular attribute for these samples are then assigned using random values that are within the minimum and maximum range of the attribute’s domain. This assignment follows a uniform distribution. Each noise dataset within the KEEL repository has been divided into five unique subsets for training and testing purposes. The quantity of testing samples for each set is presented in Table 1. The ultimate classification evaluation of each competing method is determined by calculating the average of the classification results from five divisions on each noise dataset. In addition, the majority of the datasets have a small number of samples. However, these datasets can effectively be utilized to validate the classification performance in scenarios with a small sample size.

The classical KNN algorithm is recognized for its effectiveness in scenarios with a clear separation between classes. However, accurate classification of datasets containing noise poses a greater challenge. To address this, we evaluated the proposed DC-KNN methods on selected noisy datasets from the KEEL repository. The evaluation is based on the classification accuracy as shown in Table 2.

Table 2 Classification accuracy (%) of different methods on various datasets: Values of K with the number of subgroups are presented in parenthesis

Data	KNN	Kmeans-KNN	DC-KNN1	DC-KNN2
Breast	97.66(11)	97.07(11)	97.66(11)	**98.24** (6, (7,5))
ILPD	69.71(2)	69.71(2)	68.57(7)	**70.86** (3, (3,5))
Computers	54.80(5)	57.20(1)	57.20(1)	**62.00(1, (3,4))**
ScreenType	43.73(18)	40.26(5)	43.2(7)	**44.26** (5, (5, 2, 3))
StarLightCurves	86.20(1)	88.2(1)	90.40(5)	**91.10** (1, (3,2,3))
Yeast-n	56.27(1)	56.27(1)	56.27(1)	**62.04**(2,(8, 4))
Sonar-n	78.57(2)	80.95(1)	80.95(1)	**83.33** (2, (6,3))
Iono-n	77.30(2)	84.39(2)	85.10(2)	**86.52** (1, (3,7))
Heart-n	74.07(11)	87.03(15)	87.03(8)	**88.88** (2, (4,2))
Pima-n	75.32(11)	75.64(9)	75.64(13)	**77.27** (5,(6,4))
Spambase-n	81.91(1)	78.21(14)	89.19(14)	**90.16** (1,(2,3))
Iris-n	91.66(5)	98.33(3)	**100**(7)	98.33 (1, (2, 2, 2))

The best performing method is highlighted in boldface

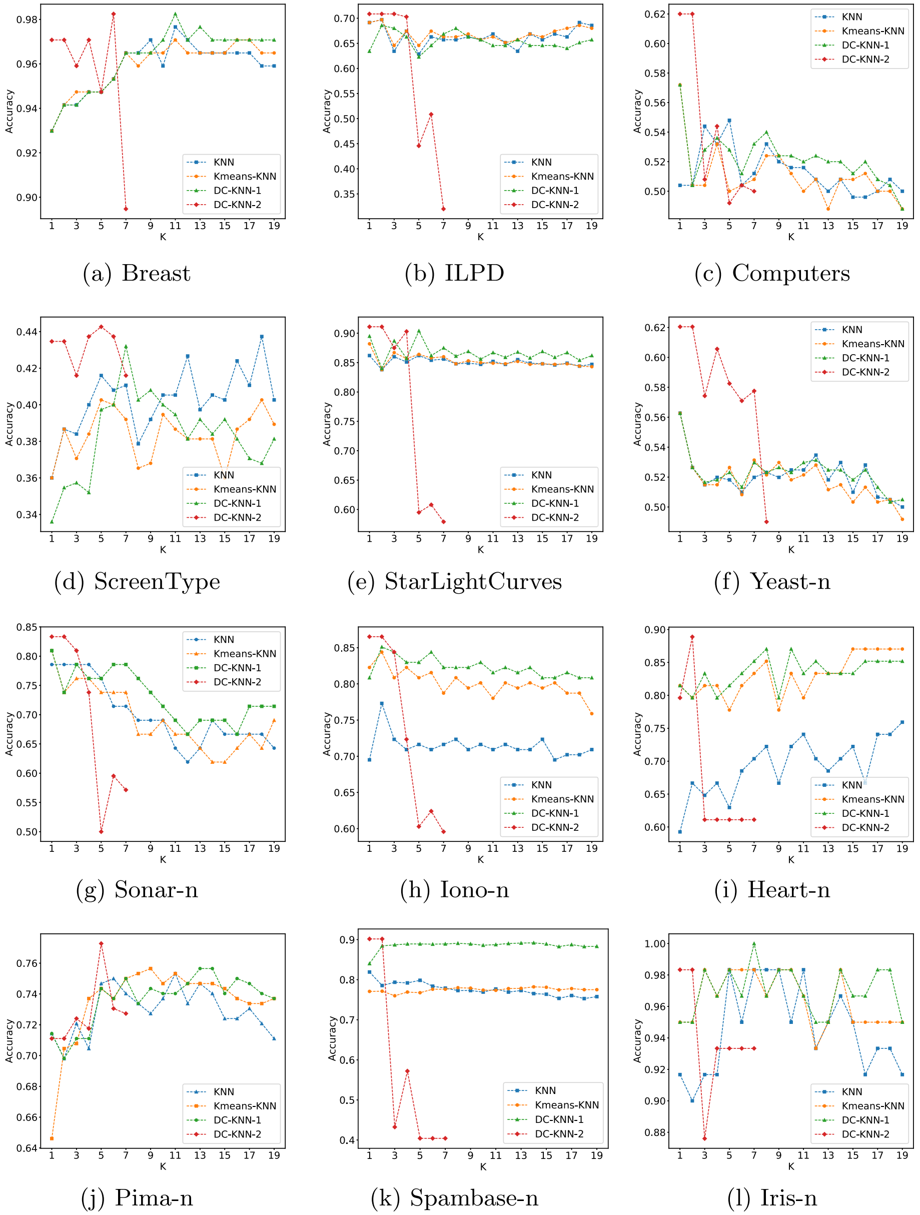
In the experiments, we assess the classification performance of the proposed DC-KNN methods by varying the neighborhood size (K) on each dataset. The parameters $(n_{c_1}, \ldots, n_{c_N})$ of the DC step, which represents the number of subgroups determined by the Silhouette method, are also taken into consideration. The values of K range from 1 to 20, and for DC-KNN2, range from 1 to 7, incrementing by 1, for all the datasets. The classification results of the suggested techniques, with different values of K, are shown in Fig. 2.

The DC-KNN2 Algorithm 3, utilizing centroids of DC for detecting the nearest neighbors, outperforms other algorithms with lower numbers of neighbors and achieves the highest level of accuracy across the majority of datasets in comparison to other algorithms.

Moreover, on most of the real data sets, the classification accuracy remains consistently constant when the value of K increases. This is because the DC-KNN2 algorithm requires that the number of neighbors be smaller than $(\min_{1 \leq i \leq N}(c_i) + 1)$. On the other hand, the DC-KNN1 Algorithm 2 consistently achieves satisfactory classification results when varying the value of K in comparison to the classical methods, particularly at higher values of K. It implies that the suggested DC-KNN1 and DC-KNN2 algorithms exhibit more robustness when the values of K are changed, while still achieving accurate classification. The explanation for this advantage may be attributed to the utilization of DC and a novel objective function in the initial phase of both approaches, which allows for the good performance of the classifier. The classification results depicted in Fig. 2 clearly show the good classification performance of the two proposed approaches. In most instances, the proposed method outperforms the other comparison methods.

4.2 Experimental Simulation Results

In order to show the effectiveness of the proposed DC-KNN classifiers, we generate and adapt different models in this subsection. In this experiment, we performed simulations using various data generating processes (DGPs) with distinct characteristics. The specifications of



(a) Breast

(b) ILPD

(c) Computers

(d) ScreenType

(e) StarLightCurves

(f) Yeast-n

(g) Sonar-n

(h) Iono-n

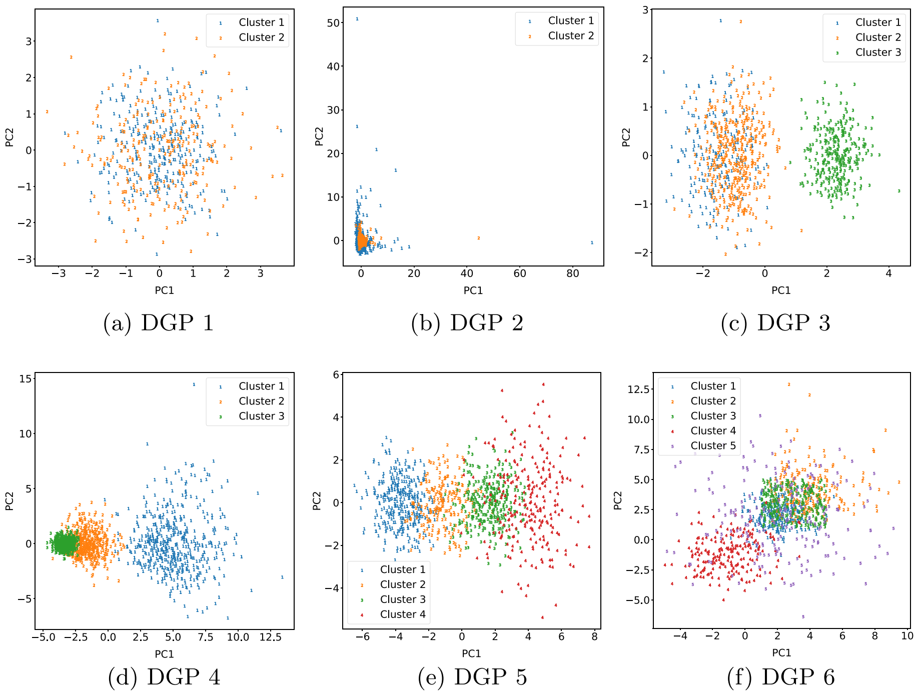
(i) Heart-n

(j) Pima-n

(k) Spambase-n

(l) Iris-n

Fig. 2 The classification accuracy of each approach is evaluated on different datasets, with varying values of K



(a) DGP 1 (b) DGP 2 (c) DGP 3

(d) DGP 4 (e) DGP 5 (f) DGP 6

Fig. 3 Two-dimensional representation using principal component analysis (PCA) simulation

the DGPs are detailed in the second section of the Supplementary Material and illustrated in Fig. 3. Specifically, we generated datasets based on different DGPs and examined scenarios where the number of clusters remained fixed at the true value, as well as scenarios where the number of clusters was estimated.

In order to determine the number of subgroups inside each apriori class, we can employ either the Silhouette or Elbow approach. Nevertheless, the traditional approaches do not ensure the enhancement of the approach's performance. Consequently, our forthcoming work will concentrate on ameliorating this aspect.

The results of the simulations performed on the data generating processes (DGPs) are displayed in Table 3. The table presents a detailed analysis of the effectiveness of the new techniques developed utilizing DC-KNN1 and DC-KNN2, which utilize the clustering results, as stated in Algorithms 2 and 3. The comparison is made over the classical KNN and Kmeans-KNN methods. As expected, utilizing the DC-KNN algorithms leads to increased accuracy values. The unsatisfactory results of classical KNN can be due to the complexity and the overlapping of data. However, a more effective approach is to first use clustering to discover hidden patterns within the classes before doing classification.

A detailed analysis of the classification effectiveness of DC-KNN techniques with varying K values can be found in Section 3 of the Supplementary Material.

The effectiveness of the proposed DC-KNN approaches in classifying diverse dataset types, including real data sets, time series data sets, noisy data sets, and simulated datasets, has been thoroughly shown through extensive experiments. Based on the results of these classification studies, it is essential to highlight important observations that emphasize the significant contributions of our research:

Table 3 Classification accuracy (%) of different methods on simulated datasets: Values of K with the number of subgroups are presented in parenthesis

Data	KNN	Kmeans-KNN	DC-KNN1	DC-KNN2
DGP1	67.27(2)	66.36(3)	67.27(4)	**68.18** (4, (5,3))
DGP2	61.5(2)	61.50(2)	62(3)	**64.00** (1, (7,3))
DGP3	76.11(2)	76.11(2)	76.11(2)	**77.22** (4, (2,5,3))
DGP4	74.66(2)	74.66(2)	74.66(2)	**85.00** (1,(2, 6, 4))
DGP5	66.11(11)	76.31(1)	76.86(3)	**78.23** (1,(4, 2, 2, 5))
DGP6	79.54(1)	82.27(2)	80.45(2)	**84.09** (3, (5, 4, 3, 3, 2))

The best performing method is highlighted in boldface

1. The DC-KNN techniques demonstrate robustness to changes in the neighborhood size K, as compared to its competitors. The experimental results consistently show that the proposed DC-KNN1 and DC-KNN2 algorithms consistently outperform other approaches and achieve good classification performance. In particular, the DC-KNN2 algorithm demonstrates strong and consistent performance when the value of K is smaller than $(\min_{1\leq i\leq N}(c_i)+1)$; this indicates a sensitivity of the DC-KNN2 algorithm to the number of centroid neighbors.
2. The process of learning clustering with adaptive distances aims to uncover concealed patterns within training groups. This is achieved by optimizing a newly proposed objective function and utilizing the outcomes of the clustering step in conjunction with the KNN classifier. This approach effectively enhances the performance of KNN-based classification.
3. DC-KNN exhibits strong performance in scenarios with limited training data. The performance of KNN-based classification can be significantly influenced by the selection of neighbors, particularly when working with various data sets and small sample sizes. Nevertheless, the experimental results in these situations demonstrate that our DC-KNN outperforms the competing methods.
4. The DC-KNN algorithms exhibit greater resilience to noisy data. Our DC-KNN algorithms outperform existing algorithms when applied to data containing noise.

The excellent performance of our methods can be attributed to several factors. Firstly, we utilize DC to uncover hidden patterns by adjusting the distances between data points. This allows us to effectively weigh the features of different subclasses. Secondly, we introduce a novel objective function that considers both the compactness within each apriori group and the separation between apriori classes. This enables us to accurately cluster the training predefined groups. Lastly, we adapt the KNN classifier to incorporate the augmented labels obtained from the clustering step, enhancing the accuracy of classification tasks. Therefore, our DC-KNN algorithms exhibit strong potential as a KNN-based classifier due to their robustness and efficacy in pattern classification.

5 Conclusions and Future Works

This research presents the DC-KNN algorithm, a novel supervised approach that combines dynamic clustering and the K-nearest neighbor classifier. The unsupervised clustering phase is used to discover new information from the original datasets that can help to improve supervised classification accuracy. DC in the unsupervised phase uses a new objective function that takes into account both intra-cluster and inter-cluster similarity, with cluster weights for variables being computed automatically and optimized as the algorithm converges. These weights can be used to identify important variables for clustering and eliminate variables that could introduce noise in the classification process. In the supervised phase, the new weights are employed to determine the nearest neighbor of new data points. Overall, the DC-KNN algorithm provides a unique and effective classification technique by combining DC and KNN.

Based on the results of the application on the real dataset test using the usual K-means and dynamical clustering algorithm to enhance the KNN supervised classification, better results were obtained than using dynamical clustering before training the supervised classifier. The reason is that the proposed method can provide more precise information and homogeneous clusters using clustering in the first step as a preprocessing step to discover the hidden patterns. As a result, the classification results obtained by the proposed method provide better results and increase the classifier's accuracy.

The focus of this study is to discover hidden information that can be comprehended through novel patterns, leading to the identification of subgroups of instances classified by these new sub-patterns within the previously established classes. The objective is to ascertain whether the implementation of the DC-KNN methodology enhances the accuracy of classification. Initially, a DC algorithm is employed to identify novel patterns within the original classes. This research demonstrates the utilization of this algorithm across a range of datasets unaffected by data points that deviate from the norm. It is worth noting that alternative metrics can be chosen to handle outliers and determine the adaptive distances between the data points and centroids.

The primary aim of this investigation is to examine the theoretical aspect of integrating unsupervised and supervised classification and to evaluate whether this novel approach enhances classification performance compared to traditional classifiers. Additionally, several techniques can be employed to ascertain the optimal number of subgroups for each original class.

In this two-stage study, the unsupervised method utilized is DC, while the supervised strategy employed is KNN. Future research aims may concentrate on the combination of different clustering techniques with alternative classifiers to investigate the performance of combining unsupervised and supervised classification using various strategies, as well as determining how such combinations can impact the final outcome. Moreover, attempts could be made to formulate an objective function that condenses the process into a single-step strategy.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00357-024-09471-5.

Funding Open access funding provided by Università degli Studi della Campania Luigi Vanvitelli within the CRUI-CARE Agreement.

Data Availability The experimental data and simulation results that support the findings of this paper are available on public repositories or upon request from the corresponding author. In particular, real datasets are sourced from the UCI Machine Learning Repository (Bache & Lichman, 2013) and the UCR Time Series Classification Repository (Dau et al., 2018). KEEL attribute noise datasets are proposed in Alcala-Fdez et al. (2011). For simulated data, we provide the data generation schema in the Supplementary Material; however, datasets are available upon request.

Declarations

Ethics Approval The research study did not involve any human participants or animals.

Conflict of Interest The authors declare no competing interests.

References

Abavisani, M., & Patel, V. M. (2019). Deep sparse representation-based classification. *IEEE Signal Processing Letters, 26*(6), 948–952.

Alayrac, J. B., Bojanowski, P., & Agrawal, N., et al. (2016). Unsupervised learning from narrated instruction videos. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 4575–4583)

Alcala-Fdez, J., Fernandez, A., & Luengo, J., et al. (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing,17*(2–3), 255–287. http://sci2s.ugr.es/KEEL/datasets.php

Bache, K., & Lichman, M. (2013). UCI machine learning repository. https://archive.ics.uci.edu/

Balzanella, A., & Verde, R. (2020). Histogram-based clustering of multiple data streams. *Knowledge and Information Systems, 62*(1), 203–238. https://doi.org/10.1007/s10115-019-01350-5

Bao, C., Peng, H., He, D., et al. (2018). Adaptive fuzzy c-means clustering algorithm for interval data type based on interval-dividing technique. *Pattern Analysis and Applications, 21*, 803–812. https://doi.org/10.1007/s10044-017-0663-2

Breiman, L. (2017). Classification and regression trees. Routledge. https://doi.org/10.1201/9781315139470

Chan, E. Y., Ching, W. K., Ng, M. K., et al. (2004). An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern recognition, 37*(5), 943–952. https://doi.org/10.1016/j.patcog.2003.11.003

Chang, J., Wang, L., & Meng G., et al. (2017). Deep adaptive image clustering. In: *Proceedings of the IEEE international conference on computer vision*, (pp. 5879–5887). https://doi.org/10.1109/ICCV.2017.626

Chen, L., Li, S., Bai, Q., et al. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing, 13*(22), 4712. https://doi.org/10.3390/rs13224712

Cherif, W. (2018). Optimization of K-NN algorithm by clustering and reliability coefficients: Application to breast-cancer diagnosis. *Procedia Computer Science, 127*, 293–299.

Chomboon, K., Chujai, P., & Teerarassamee, P., et al. (2015). An empirical study of distance metrics for k-nearest neighbor algorithm. In: *Proceedings of the 3rd international conference on industrial application engineering*

Dapas, M., Lin, F. T., Nadkarni, G. N., et al. (2020). Distinct subtypes of polycystic ovary syndrome with novel genetic associations: An unsupervised, phenotypic clustering analysis. *PLoS medicine, 17*(6), e1003132.

Dau, H. A., Keogh, E., & Kamgar, K., et al. (2018). The UCR time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018

de Carvalho, Fd. A., Irpino, A., Verde, R., et al. (2022). Batch self-organizing maps for distributional data with an automatic weighting of variables and components. *Journal of Classification, 39*(2), 343–375. https://doi.org/10.1007/s00357-022-09411-1

De Carvalho, Fd. A., & Lechevallier, Y. (2009). Partitional clustering algorithms for symbolic interval data based on single adaptive distances. *Pattern Recognition, 42*(7), 1223–1236. https://doi.org/10.1016/j.patcog.2008.11.016

Diday, E., Govaert, G., & Lechevallier, Y., et al. (1981). Clustering in pattern recognition. In: *Digital Image Processing: Proceedings of the NATO Advanced Study Institute held at Bonas, France*, June 23-July 4, 1980, (pp. 19–58). Springer

Diday, E., & Simon, J. (1976). Clustering analysis. *Digital pattern recognition*, (pp. 47–94)

Diday, E. (1971). Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques. *Revue de statistique appliquée, 19*(2), 19–33.

Diday, E., & Govaert, G. (1977). Classification automatique avec distances adaptatives. *RAIRO Informatique Computer Science, 11*(4), 329–349.

Duda, R. O., Hart, P. E., et al. (2006). *Pattern classification*. John Wiley & Sons.

Fix, E., & Hodges, J. L. (1989). Discriminatory analysis nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique,57*(3), 238–247. http://www.jstor.org/stable/1403797

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis, 38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Gou, J., Du, L., Zhang, Y., et al. (2012). A new distance-weighted k-nearest neighbor classifier. *J Inf Comput Sci, 9*(6), 1429–1436.

Gou, J., Ma, H., Ou, W., et al. (2019a). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications, 115*, 356–372. https://doi.org/10.1016/j.eswa.2018.08.021

Gou, J., Qiu, W., Yi, Z., et al. (2019b). Locality constrained representation-based k-nearest neighbor classification. *Knowledge-Based Systems, 167*, 38–52. https://doi.org/10.1016/j.knosys.2019.01.016

Gou, J., Qiu, W., Yi, Z., et al. (2019c). A local mean representation-based k-nearest neighbor classifier. *ACM Transactions on Intelligent Systems and Technology (TIST), 10*(3), 1–25. https://doi.org/10.1145/3319532

Gou, J., Sun, L., Du, L., et al. (2022). A representation coefficient-based k-nearest centroid neighbor classifier. *Expert Systems with Applications, 194*(116), 529. https://doi.org/10.1016/j.eswa.2022.116529

Irpino, A., Verde, R., & De Carvalho, Fd. A. (2014). Dynamic clustering of histogram data based on adaptive squared Wasserstein distances. *Expert Systems with Applications, 41*(7), 3351–3366. https://doi.org/10.1016/j.eswa.2013.12.001

Liao, M., Li, Y., Kianifard, F., et al. (2016). Cluster analysis and its application to healthcare claims data: A study of end-stage renal disease patients who initiated hemodialysis. *BMC nephrology, 17*(1), 1–14.

Li, C., Chen, H., Li, T., et al. (2022). A stable community detection approach for complex network based on density peak clustering and label propagation. *Applied Intelligence, 52*(2), 1188–1208.

Li, H., & Wei, M. (2020). Fuzzy clustering based on feature weights for multivariate time series. *Knowledge-Based Systems, 197*(105), 907.

Luo, S., Miao, D., Zhang, Z., et al. (2020). Non-numerical nearest neighbor classifiers with value-object hierarchical embedding. *Expert Systems with Applications, 150*(113), 206.

Malakouti, S. M. (2023). Heart disease classification based on ECG using machine learning models. *Biomedical Signal Processing and Control, 84*(104), 796.

Maturo, F., & Verde, R. (2022). Combining unsupervised and supervised learning techniques for enhancing the performance of functional data classifiers. *Computational Statistics*. https://doi.org/10.1007/s00180-022-01259-8

Pan, Z., Wang, Y., & Pan, Y. (2020). A new locally adaptive k-nearest neighbor algorithm based on discrimination class. *Knowledge-Based Systems, 204*(106), 185.

Quinlan, J. R., et al. (1996). Bagging, boosting, and c4. 5. *Aaai/Iaai, 1*, 725–730.

Rastin, N., Jahromi, M. Z., & Taheri, M. (2021). A generalized weighted distance k-nearest neighbor for multi-label problems. *Pattern Recognition, 114*(107), 526.

Rastin, N., Taheri, M., & Jahromi, M. Z. (2021). A stacking weighted k-nearest neighbour with thresholding. *Information Sciences, 571*, 605–622.

Rodríguez, S. I. R., & de Carvalho, Fd. A. T. (2022). Clustering interval-valued data with adaptive Euclidean and city-block distances. *Expert Systems with Applications, 198*(116), 774.

Ruan, Y., Xiao, Y., Hao, Z., et al. (2021). A nearest-neighbor search model for distance metric learning. *Information Sciences, 552*, 261–277.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science, 2*(3), 160.

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised k-means clustering algorithm. IEEE. *Access, 8*, 80716–80727.

Sivasankari, S., Surendiran, J., & Yuvaraj, N., et al. (2022). Classification of diabetes using multilayer perceptron. In: *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, (pp. 1–5). IEEE

Soheily-Khah, S., Marteau, P. F., & Béchet, N. (2018). Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the ISCX dataset. In: *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, (pp. 219–226). IEEE

Taunk, K., De, S., & Verma, S., et al. (2019). A brief review of nearest neighbor algorithm for learning and classification. In: *2019 international conference on intelligent computing and control systems (ICCS)*, (pp. 1255–1260). IEEE

Uddin, S., Haque, I., Lu, H., et al. (2022). Comparative performance analysis of k-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports, 12*(1), 1–11.

Wang, Z., Huang, B., & Wang, G., et al. (2023). Masked face recognition dataset and application. *IEEE Transactions on Biometrics, Behavior, and Identity Science*

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of translational medicine 4*(11)

Zhang, C., Liu, C., Zhang, X., et al. (2017a). An up-to-date comparison of state-of-the-art classification algorithms. *Expert Systems with Applications, 82*, 128–150.

Zhang, S., Li, X., Zong, M., et al. (2017b). Efficient KNN classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems, 29*(5), 1774–1785.

Zhao, Y., & Yang, L. (2023). Distance metric learning based on the class center and nearest neighbor relationship. *Neural Networks, 164*, 631–644.

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

Authors and Affiliations

Mohammed Sabri[1,2] · Rosanna Verde[2] · Antonio Balzanella[2] · Fabrizio Maturo[3] · Hamid Tairi[1] · Ali Yahyaouy[1] · Jamal Riffi[1]

✉ Mohammed Sabri
mohammed.sabri@usmba.ac.ma

✉ Antonio Balzanella
antonio.balzanella@unicampania.it

Rosanna Verde
rosanna.verde@unicampania.it

Fabrizio Maturo
fabrizio.maturo@unimercatorum.it

Hamid Tairi
hamid.tairi@usmba.ac.ma

Ali Yahyaouy
ali.yahyaouy@usmba.ac.ma

Jamal Riffi
riffi.jamal@gmail.com

[1] LISAC Laboratory, Faculty of Sciences Dhar EL Mehraz, Sidi Mohamed Ben Abdellah University, B.P. 1796 Atlas, Fez 30003, Morocco

[2] Department of Mathematics and Physics, University of Campania L. Vanvitelli, Viale Lincoln 5, Caserta 81100, Italy

[3] Faculty of Economics, Universitas Mercatorum, Piazza Mattei 10, Roma 00186, Italy