ORIGINAL ARTICLE

Unconstrained neighbor selection for minimum reconstruction error-based K-NN classifiers

Rassoul Hajizadeh[1]

Abstract

It is essential to define more convincing and applicable classifiers for small datasets. In this paper, a minimum reconstruction error-based K-nearest neighbors (K-NN) classifier is proposed. We propose a new neighbor selection method. In the proposed neighbor selection method, a subset of data that minimize the reconstruction error is assigned as the neighbors of the query sample. Also, there is not any constraint on the distance of the neighbors from the query sample. An l_0-based sparse representation problem is introduced for selecting the neighbors. These neighbors are assigned as neighbors of the minimum reconstruction error-based K-NN classifiers. Three l_0-based minimum reconstruction error-based K-NN classifiers are introduced. These classifiers are less sensitive to the reconstruction coefficients in minimum reconstruction error-based K-NN classifiers and reconstruct the query sample with less error. The results on UCI machine learning repository, UCR time-series archive datasets, and a small subset (16%) of MNIST handwritten digit database demonstrate the suitable performance of the proposed method. The recognition precision increases up to more than 3% in some evaluations.

Keywords K-NN based classifiers · Minimum reconstruction error · l_0-Based neighbor selection · Recognition rate

Introduction

Artificial intelligence and machine learning have been developed in many applications, such as face recognition [1, 2], optical character recognition (OCR) [3], medical image processing [4, 5], gesture recognition [6, 7], fault detection [8, 9], communication systems [10], and news classification [11]. The classification section is an unavoidable part of most of these applications.

It is essential to define convincing and applicable classifiers for small datasets. These days, deep learning-based classifiers obtain exciting results in different applications [12, 13]. However, deep learning-based methods need large datasets for training the network and determining parameters. Therefore, they cannot be employed on every small dataset. Already, many classifiers have been presented, such as K-nearest neighbors (K-NN) [14], support vector machine

(SVM) [15], and neural network-based classifiers [16]. K-NN based classifiers are very convenient because of their simplicity and suitable performance. In the conventional K-NN classifier, first, K-nearest neighbors of the query sample are determined based on Euclidean distance. Then, by majority voting on classes of the K selected neighbors, the query sample is classified.

Recently, some K-NN-based classifiers have been introduced such as weighted representation-based K-NN (WRKNN) [17], weighted local mean representation-based K-NN (WLMRKNN) [17], collaborative representation-based nearest neighbor (CRNN) [18], distance-weighted K-NN (DWKNN) [19], multi-local means-based nearest neighbor (MLMNN) [20], local mean-based K-NN (LMKNN) [21], pseudo-nearest neighbor (PNN) [22], local mean-based pseudo-nearest neighbor (LMPNN) [23], generalized mean-distance-based k-nearest neighbor classifier (GMDKNN) [24], and representation coefficient-based k-nearest centroid neighbor method (RCKNCN) [25]. Generally, K-NN-based classifiers can be categorized into three groups: majority voting-based, mean-distance-based, and minimum reconstruction error-based classifiers.

The majority voting-based K-NN classifiers, such as conventional K-NN, DWKNN, and CRNN, predict the category

✉ Rassoul Hajizadeh
r.hajizadeh@ausmt.ac.ir

1 Machine Learning and Deep Learning Research Laboratory, Faculty of Engineering Modern Technologies, Amol University of Special Modern Technologies, Amol, Iran

of a query sample by majority voting on the K neighbor classes. In CRNN, the query sample is linearly reconstructed versus all data with a constraint on their Euclidean distances. Then, K samples corresponding to the K largest reconstruction coefficients are selected as neighbors of the query sample. Finally, similar to the conventional K-NN, the query sample is classified by majority voting on the K neighbors classes [18]. In DWKNN, K weights are calculated according to the distances among K neighbors. Then, the query sample is assigned to the class for which the weights of representatives among the K-nearest neighbors sum to the greatest value [19].

In mean-distance-based K-NN classifiers, such as LMKNN, PNN, and LMPNN, the query sample is classified using a pseudo-neighbor that is calculated by neighbors. In LMKNN, pseudo-neighbor per class is calculated by the mean of K selected neighbors. Next, the query sample is classified to the minimum distance between the query sample and the pseudo-neighbors [21]. Similarly, in PNN, after determining K neighbors per class, a pseudo-neighbor is calculated as a weighted mean of the K neighbors per class [22]. Then, the query sample is classified into the class corresponding to the closest pseudo-neighbor. LMPNN is also an extended version of the PNN classifier, which uses local mean-based pseudo-neighbors [23]. GMDKNN uses multi-generalized mean and nested generalized mean distances, which are based on the characteristic of the generalized mean [24].

In minimum reconstruction error-based K-NN classifiers, such as MLMNN, WRKNN, and WLMRKNN, the query sample is linearly reconstructed versus the K neighbors per class. Then, the query sample is assigned to the category with the minimum reconstruction error. In MLMNN, the reconstruction coefficients are constrained by a l_2-norm on their values [20]. WRKNN calculates the coefficients with constraints on the Euclidean distance of the selected neighbors [17]. In WLMRKNN, first, local mean-based pseudo-neighbors are calculated using neighbors. Then, similar to WRKNN, the reconstruction coefficients are calculated [17].

Except for the CRNN classifier, other mentioned K-NN-based classifiers select neighbors of the query sample in the same way. While these classifiers decide the query sample with different criteria. Therefore, it can make decrease the performance of the classifiers.

Our motivation and contribution

It is necessary to introduce convincing and effective classifiers for small datasets. K-NN-based classifiers are relatively simple and efficient. In this manuscript, we try to improve their performance and increase the recognition rate. All types of K-NN-based classifiers select a subset of samples as neighbors of the query sample. The selection of neighbors is the unavoidable part of the K-NN based classifiers.

Therefore, how to choose neighbors can be pivotal to the performance of the K-NN-based classifiers. Most K-NN-based classifiers select neighbors based on the minimum Euclidean distance. The Euclidean distance-based selection of neighbors is rational for majority voting-based and mean-distance-based K-NN classifiers. However, this scheme of selection of neighbors is not logical for minimum reconstruction error-based K-NN classifiers, which decide about query sample according to the minimum error value. Sometimes, a sample is closer to a query sample than another sample, but it cannot well linearly reconstruct the query sample. It can reduce the performance of the minimum reconstruction error classifiers. On the other hand, a sample may provide minimum reconstruction error, but it is not in proximity to the query sample.

The minimum reconstruction error-based K-NN classifiers typically have the best performance [17]. In this manuscript, I propose a neighbor selection method based on the minimization reconstruction error of the query sample. In the proposed method, a subset of data that minimizes the reconstruction error is assigned as the neighbors of the query sample. Euclidian distance is not considered as a criterion to select the neighbors of the query sample. Also, an l_0-based sparse representation scheme is introduced for determining the proposed neighbors. The proposed neighbor selection method is applicable for minimum reconstruction error-based classifiers. Three l_0-MLMNN, l_0-WRKNN, and l_0-WLMRKNN classifiers are defined based on the proposed neighbor selection method.

The examples are based on different databases which include University of California Irvine (UCI) machine learning repository [26], UCR time-series classification archive [27], and a small subset of Modified National Institute of Standards and Technology (MNIST) handwritten digit database [28]. The results exhibit the suitable performance of the proposed method.

The rest of the manuscript is organized as follows. The system model and related works are presented in "System model and related works". Next, the proposed neighbor selection method and proposed K-NN-based classifiers are described. In "Simulation results", the simulation results and the discussion of the results are given. Finally, "Conclusion" concludes the manuscript.

System model and related works

Figure 1 shows the general scheme of minimum reconstruction error-based K-NN classifiers. All minimum reconstruction error-based K-NN classifiers select K samples as neighbors of the query sample per class. Then, based on the reconstruction errors, the category of the query sample is determined.
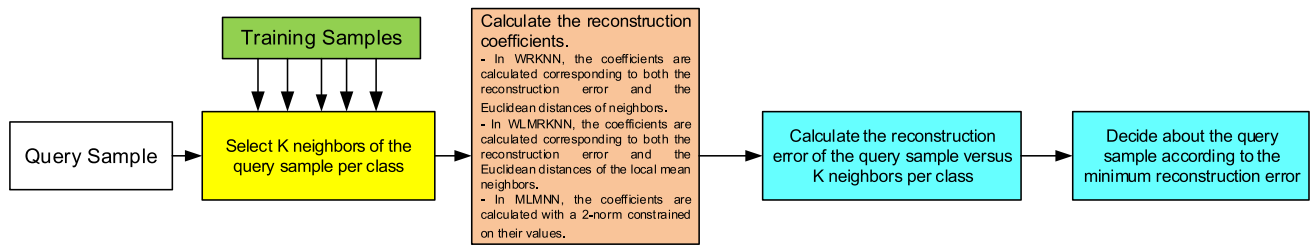
Fig. 1 The general scheme of minimum reconstruction error-based K-NN classifiers

Generally, minimum reconstruction error-based K-NN classifiers include three steps: (1) selecting neighbors using minimum Euclidean distance, (2) calculating reconstruction coefficients, and (3) classifying based on the minimum error.

Suppose $X = [x_1, \ldots, x_N] \in \mathfrak{R}^{D \times N}$ includes N labeled samples, $y \in \mathfrak{R}^D$ is the query sample, $L = \{1, \ldots, C\}$ is the label set of data (C is the number of classes), and $X_{\text{KNN}}^j = \left[x_{1\text{NN}}^j, \ldots, x_{\text{KNN}}^j \right] \in \mathfrak{R}^{D \times K}$ expresses K-nearest neighbors belong to the jth class. MLMNN, WRKNN, and WLMRKNN are presented in the following.

MLMNN classifier

First, K neighbors of the query sample are determined per class based on the minimum Euclidean distance. Then, K local mean pseudo-neighbors per class are calculated as

$$\overline{x}_{i\text{NN}}^j = \frac{1}{i} \sum_{l=1}^{i} x_{l\text{NN}}^j, \quad i = 1, \ldots, K. \tag{1}$$

Then, the query sample is linearly reconstructed per class using K local mean pseudo-neighbors with a l_2-norm constraint on reconstruction coefficients ($\boldsymbol{\beta}^j$) as

$$\boldsymbol{\beta}^{j*} = \arg \min_{\boldsymbol{\beta}^j} \left\{ \| y - \overline{X}_{\text{KNN}}^j \boldsymbol{\beta}^j \|_2^2 + \mu \| \boldsymbol{\beta}^j \|_2^2 \right\}, \tag{2}$$

where μ is a regularization parameter, $\overline{X}_{\text{KNN}}^j = \left[\overline{x}_{1\text{NN}}^j, \overline{x}_{2\text{NN}}^j, \ldots, \overline{x}_{\text{KNN}}^j \right]$, and $\boldsymbol{\beta}^j = \left[\beta_1^j, \beta_2^j, \ldots, \beta_K^j \right]$ is the reconstruction coefficients vector of the jth class. The optimum $\boldsymbol{\beta}^j$ can be calculated as a closed-form solution

$$\boldsymbol{\beta}^{j*} = \left(\left(\overline{X}_{\text{KNN}}^j \right)^{\text{T}} \overline{X}_{\text{KNN}}^j + \mu I \right)^{-1} \left(\overline{X}_{\text{KNN}}^j \right)^{\text{T}} y. \tag{3}$$

Then, reconstruction errors are computed as

$$r_{\text{MLMNN}}^j(y) = \| y - \overline{X}_{\text{KNN}}^j \boldsymbol{\beta}^{j*} \|_2^2. \tag{4}$$

Finally, the query sample is classified into the class with minimum reconstruction error [20].

WRKNN and WLMRKNN classifiers

Similar to MLMNN, first, K neighbors of the query sample are calculated per class. Then, reconstruction coefficients ($\boldsymbol{\eta}^j$) of the query sample are calculated per class with a constraint on Euclidean distances of K-nearest neighbors (X_{KNN}^j) as

$$\boldsymbol{\eta}^{j*} = \arg \min_{\boldsymbol{\eta}^j} \left\{ \| y - X_{\text{KNN}}^j \boldsymbol{\eta}^j \|_2^2 + \gamma \| T^j \boldsymbol{\eta}^j \|_2^2 \right\}, \tag{5}$$

where, γ is a regularization parameter, and T^j is a diagonal matrix of Euclidean distances as

$$T^j = \begin{bmatrix} \| y - x_{1\text{NN}}^j \|_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \| y - x_{\text{KNN}}^j \|_2 \end{bmatrix}, \tag{6}$$

and $\boldsymbol{\eta}^j$ can be solved in a closed-form per class as

$$\boldsymbol{\eta}^{j*} = \left(\left(X_{\text{KNN}}^j \right)^{\text{T}} X_{\text{KNN}}^j + \gamma \left(T^j \right)^{\text{T}} T^j \right)^{-1} \left(X_{\text{KNN}}^j \right)^{\text{T}} y. \tag{7}$$

After computing the optimum $\boldsymbol{\eta}^j$ per class, the query sample is classified to the class with minimum reconstruction error, which is calculated as

$$r_{\text{WRKNN}}^j(y) = \| y - X_{\text{KNN}}^j \boldsymbol{\eta}^{j*} \|_2^2. \tag{8}$$

Most steps of WLMRKNN are similar to WRKNN. WLMRKNN employs local mean pseudo-neighbors ($\overline{X}_{\text{KNN}}^j$, Eq. (1)) instead of nearest neighbors (X_{KNN}^j). The decision is also made based on the minimum reconstruction error of the query sample versus the pseudo-neighbors per class as

$$r_{\text{WLMRKNN}}^j(y) = \| y - \overline{X}_{\text{KNN}}^j S^{j*} \|_2^2, \quad j = 1, \ldots, C, \tag{9}$$

where S^j is the reconstruction coefficients vector of the jth class [17].

It has been shown that minimum reconstruction-based K-NN classifiers, i.e., WRKNN and WLMRKNN, have the best performance [17].
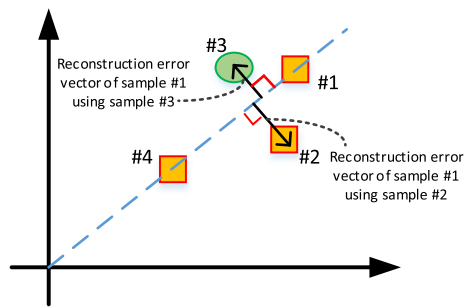
Fig. 2 An example to explain the proposed method

Proposed l_0-based neighbor selection

In minimum reconstruction error-based KNN classifiers, the neighbors are selected based on the Euclidean distance, while their decision metric is based on the minimum reconstruction error. There is no certainty that K Euclidean distance-based nearest samples obtain the minimum reconstruction error. On the other hand, it is intuitive that the samples with the same category of the query sample often provide the best representation of the query sample and minimize the reconstruction error. Therefore, choosing the neighbors based on the minimum reconstruction error can reduce the reconstruction error value and improve the accuracy of the classifiers. This statement can be justified by an example. Suppose there are four two-dimensional samples (Fig. 2).

Assign sample #1 as a query sample and three residual samples as neighbors of it. Although samples #2 and #3 are closer to the query sample than sample #4, the reconstruction error using sample #4 is less than the reconstruction error

– There is no constraint on the distance of the neighbors.
– The samples with the same category cause minimum reconstruction error.

In the proposed method, a subset of training data is selected as neighbors, which makes the minimum reconstruction error of the query sample. Also, there is not any constraint on the Euclidean distance of the neighbors. The proposed equation is defined as

$$\left\{ \boldsymbol{X_K}^*, \boldsymbol{\alpha_K}^* \right\} = \arg\min_{\boldsymbol{X_K}, \boldsymbol{\alpha_K}} \left\{ \| \boldsymbol{y} - \boldsymbol{X_K}^{\mathrm{T}} \boldsymbol{\alpha_K} \|_2 \right\}, \tag{10}$$

where $\boldsymbol{X_K}$ is a set of K samples from $\boldsymbol{X}$, and $\boldsymbol{\alpha_K} \in \mathfrak{R}^{K \times 1}$ is the vector of coefficients. Equation (10) can be rewritten as

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \left\{ \| \boldsymbol{y} - \boldsymbol{X}^{\mathrm{T}} \boldsymbol{\alpha} \|_2 \right\}, \quad \text{s.t. } \|\boldsymbol{\alpha}\|_0 = K, \tag{11}$$

where $\boldsymbol{\alpha} \in \mathfrak{R}^{N \times 1}$ is a sparse vector of coefficients with K nonzero values, and $\boldsymbol{X_K}^*$ is determined from $\boldsymbol{X}$ according to the nonzero values of $\boldsymbol{\alpha}^*$. Equation (11) is an l_0-based sparse representation problem. Equation (11) is an NP-hard problem, and the orthogonal matching pursuit method is a semi-optimum method to solve it. Suppose $\boldsymbol{X_{K-l_0}} = \left[\boldsymbol{x_{1-l_0}}, \boldsymbol{x_{2-l_0}}, \ldots, \boldsymbol{x_{K-l_0}} \right]$ is the set of K l_0-based selected neighbors, K is the number of neighbors, $\boldsymbol{y}$ is the query sample, and $\varnothing$ is the symbol of an empty set. Algorithm 1 shows steps of the proposed l_0-based neighbor selection method. $\boldsymbol{X_{K-l_0}}$ includes K samples that reconstruct $\boldsymbol{y}$ with minimum error.

Algorithm 1: *l_0-based neighbor selecting*

Input: labeled samples ($\boldsymbol{X}$), the number of neighbors (K), and the query sample ($\boldsymbol{y}$),

Output: l_0-based neighbors $\left(\boldsymbol{X_{K-l_0}} \right)$,

---

Initialization: $\boldsymbol{r} = \boldsymbol{y}$, $\boldsymbol{X_{K-l_0}} = \emptyset$;

- normalization of $\boldsymbol{X}$: $\boldsymbol{X_n} = \left\{ \frac{\boldsymbol{x_i}}{\|\boldsymbol{x_i}\|_2}, i = 1, \ldots, N \right\}$

for i =1 to K do

1. $i^* = \arg\max_i \left\{ \left| \langle \boldsymbol{r}, \boldsymbol{x_{i_n}} \rangle \right|, i = 1, \ldots, N \right\}$
2. select $\boldsymbol{x_{i^*}}$ as neighbor and insert it to $\boldsymbol{X_{K-l_0}}$
3. remove the i^*th sample from $\boldsymbol{X}$ and $\boldsymbol{X_n}$,
4. calculate and update $\boldsymbol{\alpha}$: $\boldsymbol{\alpha} = \left(\boldsymbol{X_{K-l_0}}^T \boldsymbol{X_{K-l_0}} \right)^{-1} \boldsymbol{X_{K-l_0}}^T \boldsymbol{y}$,
5. update $\boldsymbol{r}$: $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X_{K-l_0}} * \boldsymbol{\alpha}$,

end for

---

using samples #2 and #3. In fact, based on the minimum reconstruction error criterion, sample #4 can better represent the query sample. In this manuscript, a neighbor selection method is proposed based on the below principles:

In the proposed method, the K samples are selected as neighbors that obtain minimum reconstruction error. There is no constraint on Euclidean distances of the neighbors, and each K subset of samples can be assigned as neighbors. Therefore, the reconstruction error value using the proposed

method is less than the reconstruction error value using the neighbors that are based on the minimum Euclidean distance. Figure 3 shows the diagram of the proposed neighbor selection method.

In the following, K selected samples are assigned as neighbors of minimum reconstruction error-based K-NN classifiers. Based on the proposed neighbor selection method, l_0-MLMNN, l_0-WRKNN, and l_0-WLMRKNN are introduced in Algorithms 2–4, respectively.

In WRKNN and WLMRKNN, neighbors are close to the query sample. However, in the proposed l_0-WRKNN and l_0-WLMRKNN, there is not any constraint on the distance of the neighbors from the query sample, and it can vary in a wide range. Therefore, we normalize the Euclidean distance matrix in l_0-WRKNN (step 3) and l_0-WLMRKNN (step 4).

Algorithm 2: *l_0-MLMNN classifier*

Input: training samples ($\boldsymbol{X}$), the number of neighbors (K), and the query sample ($\boldsymbol{y}$),

Output: predicting the class of the query sample ($c_{\boldsymbol{y}}$),

Initialization: regularization parameter μ;

1. calculate K l_0-based neighbors of $\boldsymbol{y}$ per class using Algorithm 1, i.e., $\boldsymbol{X}_{K-l_0}^j = [\boldsymbol{x}_{1-l_0}^j, \boldsymbol{x}_{2-l_0}^j, \dots, \boldsymbol{x}_{K-l_0}^j]$,
2. compute K local mean pseudo neighbors per class, i.e., $\overline{\boldsymbol{x}}_{i-l_0}^j = \frac{1}{i}\sum_{l=1}^i \boldsymbol{x}_{l-l_0}^j , i = 1, \dots, K$.
3. calculate $\boldsymbol{\beta}_{K-l_0}^j$ per class, i.e., $\boldsymbol{\beta}_{K-l_0}^{j^*} = \left(\left(\overline{\boldsymbol{X}}_{K-l_0}^j\right)^T \overline{\boldsymbol{X}}_{K-l_0}^j + \mu\boldsymbol{I}\right)^{-1} \left(\overline{\boldsymbol{X}}_{K-l_0}^j\right)^T \boldsymbol{y}$,
4. compute the reconstruction error per class $r_{l_0-MLMNN}^j(\boldsymbol{y}) = \left\|\boldsymbol{y} - \overline{\boldsymbol{X}}_{K-l_0}^j \boldsymbol{\beta}_{K-l_0}^{j^*}\right\|_2^2$,
5. $c_{\boldsymbol{y}} = arg\min_j\{r_{l_0-MLMNN}^j(\boldsymbol{y})\}, j = 1, \dots, C$.

Algorithm 3: *l_0-WRKNN classifier*

Input: training samples ($\boldsymbol{X}$), the number of neighbors (K), and the query sample ($\boldsymbol{y}$),

Output: predicting the class of the query sample ($c_{\boldsymbol{y}}$),

Initialization: regularization parameter γ;

1. Calculate K l_0-based neighbors of $\boldsymbol{y}$ per class using Algorithm 1, i.e., $\boldsymbol{X}_{K-l_0}^j = [\boldsymbol{x}_{1-l_0}^j, \boldsymbol{x}_{2-l_0}^j, \dots, \boldsymbol{x}_{K-l_0}^j]$,
2. compute distance matrix $\boldsymbol{T}_{K-l_0}^j = diag\left(\left\|\boldsymbol{y} - \boldsymbol{x}_{1-l_0}^j\right\|_2, \dots, \left\|\boldsymbol{y} - \boldsymbol{x}_{K-l_0}^j\right\|_2\right)$,
3. normalize distance matrix $\boldsymbol{T}_{K-l_0}^j$: $\boldsymbol{T}_{K-l_0}^j = \frac{\boldsymbol{T}_{K-l_0}^j}{max\left(\left\|\boldsymbol{y}-\boldsymbol{x}_{1-l_0}^j\right\|_2, \dots, \left\|\boldsymbol{y}-\boldsymbol{x}_{K-l_0}^j\right\|_2\right)}$
4. calculate $\boldsymbol{\eta}_{K-l_0}^j$ per class, i.e., $\boldsymbol{\eta}_{K-l_0}^{j^*} = \left(\left(\boldsymbol{X}_{K-l_0}^j\right)^T \boldsymbol{X}_{K-l_0}^j + \gamma\left(\boldsymbol{T}_{K-l_0}^j\right)^T \boldsymbol{T}_{K-l_0}^j\right)^{-1} \left(\boldsymbol{X}_{K-l_0}^j\right)^T \boldsymbol{y}$,
5. compute the reconstruction errors per class $r_{l_0-WRKNN}^j(\boldsymbol{y}) = \left\|\boldsymbol{y} - \boldsymbol{X}_{K-l_0}^j \boldsymbol{\eta}_{K-l_0}^{j^*}\right\|_2^2$,
6. $c_{\boldsymbol{y}} = arg\min_j\{r_{l_0-WRKNN}^j(\boldsymbol{y})\}, j = 1, \dots, C$.

Algorithm 4: *l_0-WLMRKNN classifier*

Input: training samples ($\boldsymbol{X}$), the number of neighbors (k), and the query sample ($\boldsymbol{y}$),

Output: predicting the class of the query sample ($c_{\boldsymbol{y}}$),

Initialization: regularization parameter δ;

1. Calculate K l_0-based neighbors of $\boldsymbol{y}$ per class using algorithm 1, i.e., $\boldsymbol{X}_{K-l_0}^j = [\boldsymbol{x}_{1-l_0}^j, \boldsymbol{x}_{2-l_0}^j, \dots, \boldsymbol{x}_{K-l_0}^j]$,
2. compute K local mean pseudo neighbors per class, i.e., $\overline{\boldsymbol{x}}_{i-l_0}^j = \frac{1}{i}\sum_{l=1}^i \boldsymbol{x}_{l-l_0}^j , i = 1, \dots, K$.
3. compute distance matrix $\overline{\boldsymbol{T}}_{K-l_0}^j = diag\left(\left\|\boldsymbol{y} - \overline{\boldsymbol{x}}_{1-l_0}^j\right\|_2, \dots, \left\|\boldsymbol{y} - \overline{\boldsymbol{x}}_{K-l_0}^j\right\|_2\right)$,
4. normalize distance matrix $\overline{\boldsymbol{T}}_{K-l_0}^j$: $\overline{\boldsymbol{T}}_{K-l_0}^j = \frac{\overline{\boldsymbol{T}}_{K-l_0}^j}{max\left(\left\|\boldsymbol{y}-\overline{\boldsymbol{x}}_{1-l_0}^j\right\|_2, \dots, \left\|\boldsymbol{y}-\overline{\boldsymbol{x}}_{K-l_0}^j\right\|_2\right)}$
5. calculate $\overline{\boldsymbol{\eta}}_{K-l_0}^j$, i.e., $\overline{\boldsymbol{\eta}}_{K-l_0}^{j^*} = \left(\left(\overline{\boldsymbol{X}}_{K-l_0}^j\right)^T \overline{\boldsymbol{X}}_{K-l_0}^j + \delta\left(\overline{\boldsymbol{T}}_{K-l_0}^j\right)^T \overline{\boldsymbol{T}}_{K-l_0}^j\right)^{-1} \left(\overline{\boldsymbol{X}}_{K-l_0}^j\right)^T \boldsymbol{y}$,
6. compute the reconstruction errors per class $r_{l_0-WLMRKNN}^j(\boldsymbol{y}) = \left\|\boldsymbol{y} - \overline{\boldsymbol{X}}_{K-l_0}^j \overline{\boldsymbol{\eta}}_{K-l_0}^{j^*}\right\|_2^2$,
7. $c_{\boldsymbol{y}} = arg\min_j\{r_{l_0-WLMRKNN}^j(\boldsymbol{y})\}, j = 1, \dots, C$.

Fig. 3 The diagram of the proposed l_0-based neighbor selection method

Table 1 The characteristics of the seven UCI datasets [26]

Dataset	Number of samples	Attribute	Category
Balance	625	4	3
Climate	360	18	2
Parkinson's	195	22	2
Seeds	210	7	3
Sonar	208	60	2
Vowel	528	10	11
Wine	178	13	3

Also, MLMNN, WRKNN, and WLMRKNN classifiers consist of two nested loops: a loop with C repetitions (C is the number of categories) and a loop for determining K neighbors with $O(N)$. Therefore, the computational complexity of the MLMNN, WRKNN, and WLMRKNN is $O(C \times N)$. In the other word, the proposed l_0-MLMNN, l_0-WRKNN, and l_0-WLMRKNN classifiers consist of three nested loops: a loop with C repetitions (C is the number of categories) and two nested loops with $O(N \times K)$ for determining the l_0-based neighbors. Therefore, the computational complexity of the proposed l_0-based classifiers is $O(C \times N \times K)$.

Simulation results

Performance of the proposed l_0-based neighbor selection method is investigated on UCI machine learning repository, UCR time-series classification archive, and a small subset of MNIST handwritten digit database. In [17], it has been shown that the minimum reconstruction error-based K-NN classifiers have the best performance among K-NN-based classifiers. It is shown that the proposed l_0-based neighbor selection method improves the performance of the minimum reconstruction error-based K-NN classifiers and increases the precision of the classifiers. The regularization parameters are set as $\mu = \gamma = \delta = 0.5$. Also, the results are compared with the SVM classifier on experimented databases.

The results of the evaluation on UCI and UCR datasets

The proposed method is evaluated on seven datasets of the UCI machine learning repository and five datasets of the UCR time-series classification archive. Characteristics of the employed UCI and UCR datasets are given in Tables 1 and 2, respectively.

Each of the UCI datasets is randomly divided into the training (66.7%) and test (33.3%) subsets. The recognition rates of each UCI dataset are provided in each K by averaging the results for 50 independent iterations. Then, the

In the following, the computational complexity of the proposed l_0-based neighbor selection method and proposed classifiers are investigated. Generally, determining K-nearest data points to the query sample is the same in all K-NN-based classifiers. Based on the brute-force neighbor search, the time complexity of the K-NN-based classifiers is $O(N)$. The proposed method (Algorithm 1) consists of two nested loops: a loop with K repetitions (for determining the K neighbors) and a loop with N repetitions (for doing step 1 of the algorithm). Therefore, the l_0-based neighbor selection increases the computational complexity. The computational complexity of the proposed method is $O(N \times K)$, while the computational complexity of the Euclidean distance-based neighbor selection is $O(N)$.

Table 2 The characteristics of the five UCR time-series datasets [27]

Dataset	Number of training samples	Number of query samples	Category	Time-series length (attribute)
Chlorine concentration	467	3840	3	166
CinC_ECG_Torso	40	1380	4	1639
Fish	175	175	7	463
ItalyPowerDemand	67	1029	2	24
Non-invasive fetal ECG Thorax1	1800	1965	42	750
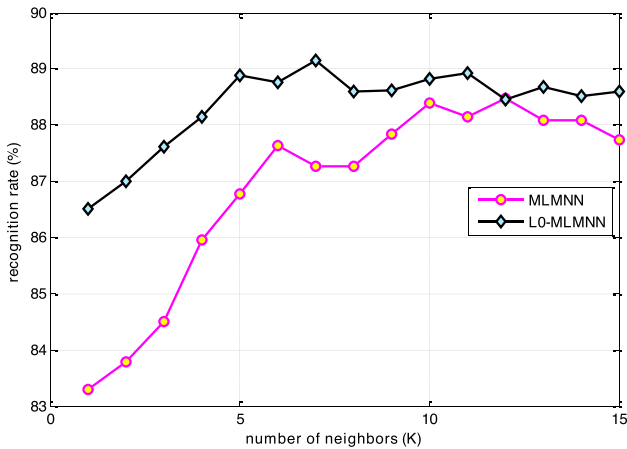
Fig. 4 The mean recognition rates on all seven UCI datasets using l_0-MLMNN and MLMNN classifiers
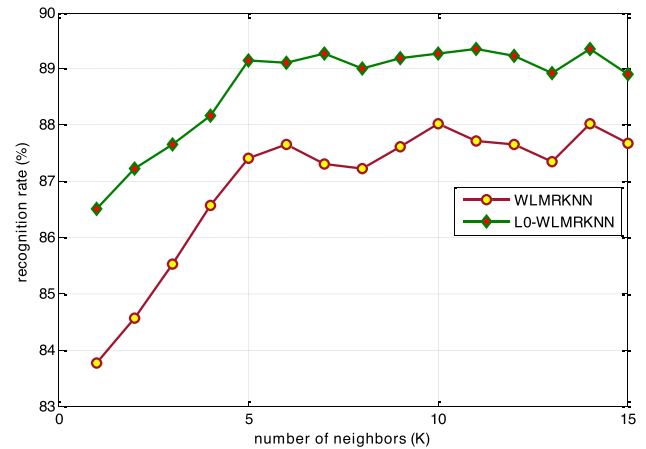
Fig. 6 The mean recognition rates on all seven UCI datasets using l_0-WLMRKNN and WLMRKNN classifiers
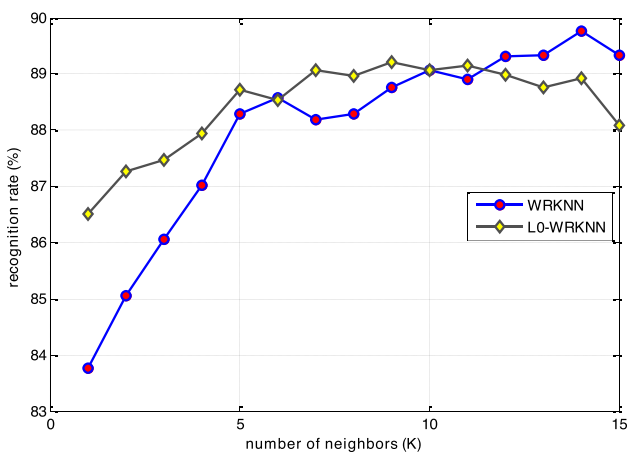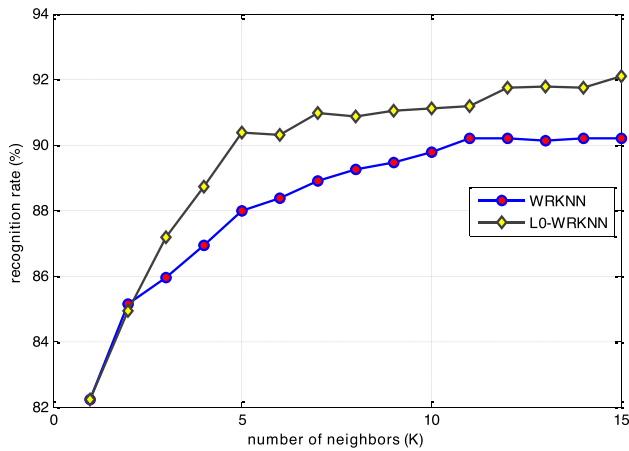

Fig. 5 The mean recognition rates on all seven UCI datasets using l_0-WRKNN and WRKNN classifiers
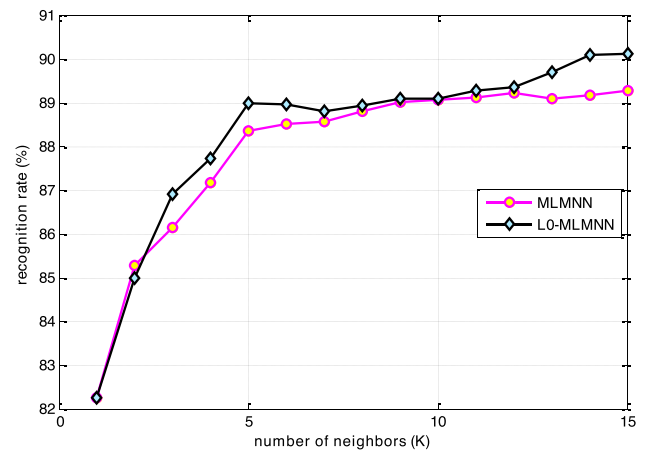
Fig. 7 The mean recognition rates on the five UCR datasets using l_0-MLMNN and MLMNN classifiers

average of the recognition rate on seven datasets is calculated for $K = 1,\ldots,15$. Figures 4, 5 and 6 show the mean of the recognition rates on the seven UCI datasets. Also, the standard-deviation values of the accuracy on seven UCI datasets are given for different numbers of the neighbors in

Table 3. The standard-deviation values of the proposed classifiers are less than the investigated minimum reconstruction error-based KNN classifiers for most numbers of neighbors. The accuracy and standard-deviation values show that the proposed l_0-based method almost improves the performance of classifiers on all seven UCI datasets.

Table 3 The standard-deviation values of the accuracy for different numbers of the neighbors using the proposed classifiers on seven datasets of UCI

Classifiers	MLMNN	L0MLMNN	WRKNN	L0WRKNN	WLMRKNN	L0WLMRKNN
$K = 1$	8.21	7.28	7.71	7.28	7.71	7.28
$K = 2$	8.52	7.84	7.53	7.93	7.47	7.96
$K = 3$	8.57	8.97	7.24	8.89	7.31	9.04
$K = 4$	8.28	7.64	6.70	8.02	6.76	7.69
$K = 5$	7.89	7.46	6.81	7.39	6.75	6.98
$K = 6$	8.11	7.10	7.05	6.21	7.21	6.76
$K = 7$	7.89	7.33	6.91	5.82	7.01	6.70
$K = 8$	7.58	7.08	6.88	4.95	6.94	6.25
$K = 9$	7.42	7.25	6.81	4.57	6.43	6.42
$K = 10$	7.53	7.04	6.96	4.84	6.55	6.20
$K = 11$	7.38	6.88	6.50	4.05	6.43	6.13
$K = 12$	7.21	7.01	6.49	4.23	6.73	6.27
$K = 13$	7.13	6.80	6.23	4.27	6.77	5.68
$K = 14$	6.71	6.55	6.26	4.01	6.86	5.70
$K = 15$	7.47	6.82	6.22	3.77	6.52	5.70

Fig. 8 The mean recognition rates on the five UCR datasets using l_0-WRKNN and WRKNN classifiers
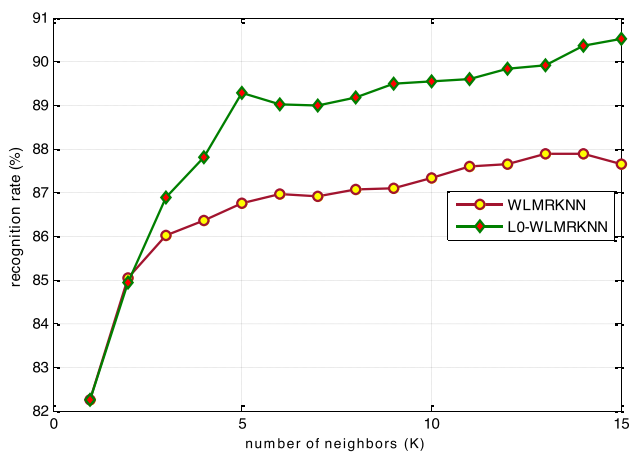
Fig. 9 The mean recognition rates on the five UCR datasets using l_0-WLMRKNN and WLMRKNN classifiers

UCR includes some time-series datasets. In UCR, the recognition rates are calculated in accordance with the given test and training subsets of each dataset. Then, the average of the recognition rate on five datasets are calculated for $K = 1,\ldots, 15$. Figures 7, 8 and 9 show the mean recognition rates on five UCR datasets using three common and proposed l_0-based minimum reconstruction error-based K-NN classifiers. Also, Table 4 shows the standard-deviation values of the accuracy for different numbers of the neighbors on five UCR datasets. The higher recognition rate values and lower standard-deviation values demonstrate that the proposed method improves the performance of the minimum reconstruction error-based K-NN classifiers on all five UCR datasets.

As a consideration, Fig. 5 shows the performance of the l_0-WRKNN is worse than WRKNN when the number of neighbors is greater than 12. In the WRKNN classifier, the distances among the query sample and neighbors are influential in calculating the coefficients vector; and then on the reconstruction error value (Eqs. 5 and 6). In the proposed l_0-based neighbor selection method, when the number of neighbors increases, the samples with high Euclidean distances from the query sample can be selected as neighbors, because there is no constraint on the Euclidean distance of the neighbors. Therefore, the neighbors with extremely high Euclidean distances are lower effective in reconstructing the query sample. On the other hand, increasing the number of neighbors provides more freedom to reconstruct the query sample. Consequently, the WRKNN classifier sometimes performs better than the l_0-WRKNN for higher numbers of neighbors. However, the WLMRKNN classifier is similar to WRKNN. However, in WLMRKNN, the local mean-based

Table 4 The standard-deviation values of the accuracy for different numbers of the neighbors using the proposed classifiers on five datasets of UCR

Classifiers	MLMNN	L0MLMNN	WRKNN	L0WRKNN	WLMRKNN	L0WLMRKNN
$K = 1$	11.76	11.76	11.76	11.76	11.76	11.76
$K = 2$	10.18	10.01	9.99	10.08	10.07	10.08
$K = 3$	9.64	8.68	9.29	8.00	9.40	8.70
$K = 4$	9.52	8.32	8.77	6.66	9.23	8.23
$K = 5$	9.69	8.66	8.45	6.53	8.99	8.42
$K = 6$	9.64	8.63	7.88	6.68	8.88	8.43
$K = 7$	9.45	8.73	7.42	5.93	8.77	8.38
$K = 8$	9.18	8.56	7.13	6.07	8.55	8.18
$K = 9$	9.04	8.50	6.74	5.91	8.36	8.02
$K = 10$	8.80	8.43	6.62	5.77	8.23	7.94
$K = 11$	8.67	8.31	6.36	5.71	8.13	7.90
$K = 12$	8.63	8.22	6.32	4.92	8.11	7.72
$K = 13$	8.67	7.84	6.32	4.89	7.90	7.61
$K = 14$	8.64	7.56	6.17	4.87	7.86	7.15
$K = 15$	8.52	7.54	6.21	4.37	8.01	6.97

neighbors (Eq. 1) have been used to reconstruct the query sample. The Euclidean distances of the local mean-based neighbors are not very high because of the mean operation on the pre-selected neighbors. Thus, increasing the number of neighbors does not reduce the performance of the l_0-WLMRKNN (Figs. 6 and 9).

In addition, McNemar's statistical test is used to compare the proposed l_0-based classifiers and the mentioned minimum reconstruction error-based K-NN classifiers. McNemar's test is a statistical method to compare the performance of two classifiers on the same test set. Suppose there are two classifiers: classifier A and classifier B. In McNemar's test, the null hypothesis is defined as A and B classifiers having the same error rate (i.e., $n_{01} = n_{10}$). Thus, the alternative hypothesis is that the performances of the classifiers are not the same. The below parameters are considered for either A and B classifiers:

– n_{01}: number of test samples misclassified by A but not by B,
– n_{10}: number of test samples misclassified by B but not by A.

Then, χ^2 statistic value is computed as $\chi^2 = (n_{01} - n_{10})^2/(n_{01} + n_{10})$. χ^2 is a chi-squared distribution with one degree of freedom. For a significance threshold of 0.05, i.e., p-value = 0.0.5, if the χ^2 statistic value is greater than 3.48, the null hypothesis is rejected, and there is a significant difference between the A and B classifiers. In the following, the results of McNemar's test (χ^2 statistic value) on five UCR datasets are given for different numbers of the neighbors in Table 5.

The results demonstrate that there is a significant difference; and the recognition rate values (Figs. 7, 8 and 9) show the superiority of the proposed l_0-based classifiers, especially l_0-WLMRKNN, at most numbers of neighbors on most investigated UCR datasets. Of course, it should be noted that the results of McNemar's test have not been provided on UCI datasets, because, in the evaluated experiments, the recognition rates of each UCI dataset are provided in each K by averaging the results for 50 independent iterations. Also, each UCI dataset is randomly divided into the training and test subsets in each iteration.

The results of the evaluation on MNIST handwritten digit database

In this manuscript, a small subset of MNIST is used to evaluate the proposed method. MNIST database includes 60,000 train and 10,000 test samples of English handwritten digit images [28]. A training subset with 10,000 samples and a test subset with 5000 samples are used in our experiments. The train and test subsets are randomly selected from the train and test samples, respectively. The recognition rates using l_0-MLMNN, l_0-WRKNN, and l_0-WLMRKNN classifiers are given in Figs. 10, 11 and 12, respectively. The results demonstrate that the proposed method improves the performance of all three reconstruction error-based KNN classifiers. Also, the results of the McNemar statistical test on the MNIST dataset are given for different numbers of the neighbors in Table 6. The results demonstrate that there is a significant difference between the investigated paired classifiers, especially for the number of neighbors of less than 9. Again, the

Table 5 The results of McNemar's test (χ^2 statistic value) of three paired MLMNN and l_0-MLMNN, WRKNN and l_0-WRKNN, and WLMRKNN and l_0-WLMRKNN classifiers for different numbers of the neighbors on five UCR datasets

UCR datasets	Number of neighbors (K) / Paired classifiers	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Chlorine concentration	MLMNN and L0MLMNN	0	0.74	**16.86**	**25.8**	**29.01**	**38.21**	**31.68**	**24.47**	**19.41**	**16.52**	**15.84**	**17.38**	**27.63**	**28.52**	**23.82**
	WRKNN and L0WRKNN	0	0.2	**30.3**	**89.8**	**198.4**	**220.8**	**207.4**	**184.5**	**154.7**	**148.5**	**126.0**	**113.8**	**124.9**	**123.2**	**126.4**
	WLMRKNN and L0WLMRKNN	0	0.02	**10.25**	**22.07**	**40.52**	**52.22**	**53.60**	**51.80**	**46.46**	**45.91**	**45.25**	**50.94**	**46.78**	**59.16**	**67.85**
CinC_ECG_Torso	MLMNN and L0MLMNN	0	0.05	**4.24**	**6.82**	2.33	2.88	3.56	3.27	**4.5**	**4**	0.33	2	3	3	3
	WRKNN and L0WRKNN	0	0.2	2.13	0.57	**18.67**	**19.56**	**17.82**	**20.16**	**18.24**	**9.85**	**11.56**	**9**	**7.35**	**7.35**	**7.35**
	WLMRKNN and L0WLMRKNN	0	0.80	0.67	**4.12**	**50.90**	**51.2**	**56.82**	**59.85**	**71.05**	**66.46**	**63.48**	**63.23**	**60.5**	**60.5**	**60.5**
Fish	MLMNN and L0MLMNN	0	0.47	0.08	0	0.33	2	2.27	2.27	2.78	2.78	2	2	0.14	0.4	0.4
	WRKNN and L0WRKNN	0	0.05	1	1.6	0	2	0.5	1	1.8	1.8	2.67	0.5	0.5	0.14	2.78
	WLMRKNN and L0WLMRKNN	0	0.05	0.4	1.6	2	0	0	0	0.5	0.14	0	0.11	0	1.29	3.6
ItalyPowerDemand	MLMNN and L0MLMNN	0	0.33	0.11	0.4	1.33	0.69	0.69	0.33	0.33	1.67	2.25	2.25	2.25	1.67	2.25
	WRKNN and L0WRKNN	0	0.33	0.4	0.82	3	**4.5**	**6**	**8**	**9**	**6.23**	2	1.29	2.78	1.29	0
	WLMRKNN and L0WLMRKNN	0	0.09	0.5	0.4	1.33	0.33	0.09	0.11	1	0.5	0.5	1	0.4	0.4	0.11
Non-invasive fetal ECG Thorax1	MLMNN and L0MLMNN	0	0.01	**7.45**	1.78	**7.35**	**7.68**	**6.40**	**6.86**	**6.05**	2.33	0.93	0.60	1.03	1.03	0.32
	WRKNN and L0WRKNN	0	0.06	**5.85**	1.11	**6**	**7.08**	**6.63**	0.64	**3.97**	1.53	0.44	0.32	0.84	2.06	1.14
	WLMRKNN and L0WLMRKNN	0	0.06	**5.48**	**4.48**	**11.46**	**8.58**	**8.91**	**11.13**	**10.8**	**7.67**	**6.4**	**7.35**	**10**	**10.56**	**10.33**

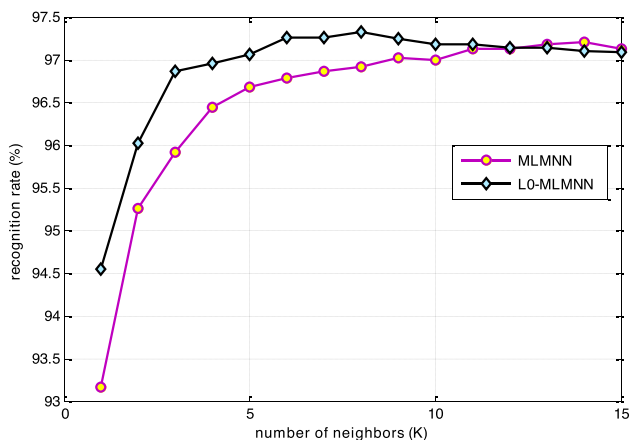McNemar's statistic values greater than 3.84 are bolded

Fig. 10 The recognition results on the MNIST database using l_0-MLMNN and MLMNN classifiers
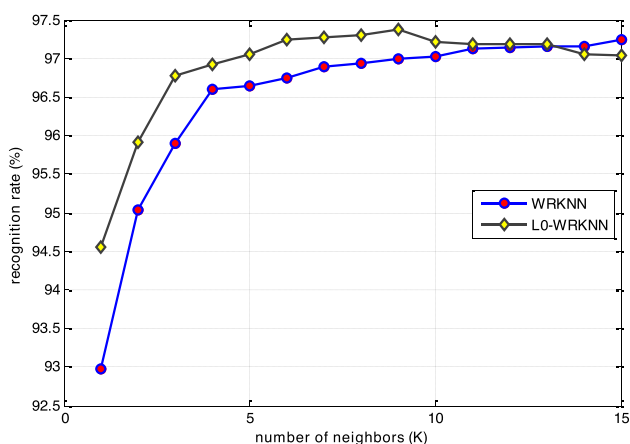

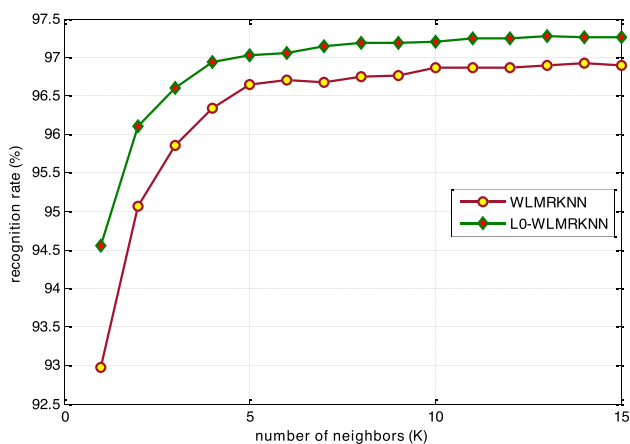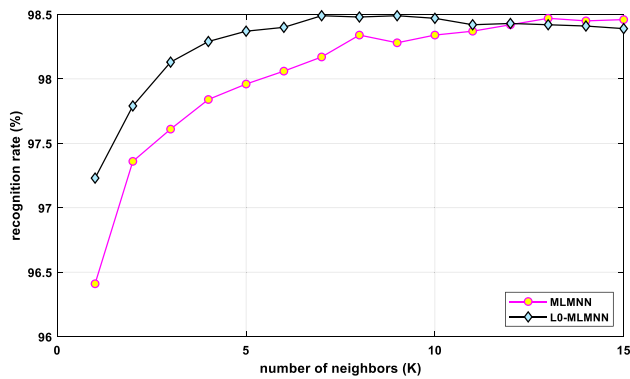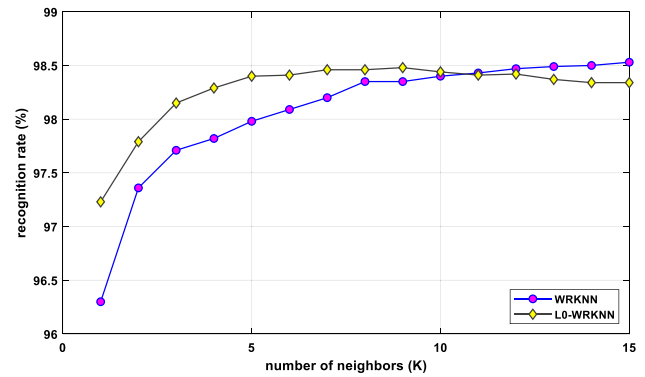**Fig. 11** The recognition results on the MNIST database using l_0-WRKNN and WRKNN classifiers

Fig. 12 The recognition rates on the MNIST database using l_0-WLMRKNN and WLMRKNN classifiers

Table 6 The results of McNemar's test (χ^2 statistic value) of three paired MLMNN and l_0-MLMNN, WRKNN and l_0-WRKNN, and WLMRKNN and l_0-WLMRKNN classifiers for different numbers of the neighbors on MNIST dataset

MNIST dataset	Number of neighbors (K)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Paired classifiers															
	MLMNN and L0MLMNN	**38.09**	**9.38**	**21.33**	**7.02**	**5.26**	**7.58**	**5.58**	**6.54**	2.45	2.53	0.11	0.01	0.01	0.05	0.05
	WRKNN and L0WRKNN	**47.89**	**19.70**	**21.74**	**3.97**	**6.7**	**6.53**	**6.05**	**5.41**	2.04	1.99	0.95	0.05	0.01	0.33	1.39
	WLMRKNN and L0WLMRKNN	**47.89**	**16.11**	**22.62**	**9.91**	**6.05**	**10.65**	**12.19**	**11.54**	**7.02**	**4.25**	3.6	2.28	1.67	0.93	1.14

McNemar's statistic values greater than 3.84 are bolded

Fig. 13 The recognition results on the whole MNIST database using l_0–MLMNN and MLMNN classifiers

Fig. 14 The recognition results on the whole MNIST database using l_0-WRKNNNNN and WRKNN classifiers

Fig. 15 The recognition results on the whole MNIST database using l_0-WLMRKNN and WLMRKNN classifiers

recognition rate values show the proposed l_0-based classifiers, especially l_0-WLMRKNN, have the best performance.

Figures 10 and 11 show that the proposed method has similar recognition rates to the conventional minimum reconstruction error-based KNN classifiers for large K. The larger numbers of neighbors cause larger samples to be involved in reconstructing the query sample. Therefore, there is more freedom in reconstructing the query sample, and it can cause the performance of the classifiers will be closer to each other. However, the performance is depended on the data distribution and classifier type.

Also, as is inferrable, the proposed l_0-based neighbor selection method can be more effective on datasets with a small number of samples or datasets with high variability per class. Results on MNIST show that the proposed l_0-based classifiers and the mentioned minimum reconstruction error-based K-NN classifiers perform the same for the large amount of K when the training subsets include more than 10,000 samples. K-nearest neighbors of the query sample likely make minimum reconstruction error when there are datasets with large amounts of samples per class.

The results on the whole MNIST digit database are evaluated for more investigations in the following. The recognition rate results are given in Figs. 13, 14 and 15. The results are similar to the obtained results on a small subset of MNIST (Figs. 10, 11 and 12) and demonstrate the better performance of the proposed method, especially for the smaller number of neighbors. For the number of neighbors greater than 12, the performance of the WRKNN is better than the l_0-WRKNN, but l_0-WLMRKNN and l_0-MLMNN perform similarly to WLMRKNN and MLMNN, respectively. The reasons have been described in the paragraph above Table 4. Also, in the MLMNN, the same as WLMRKNN, the local mean-based neighbors have been used to reconstruct the query sample. Therefore, the high Euclidean distances of the l_0-based neighbors do not have any unsuitable effect on calculating and controlling the reconstruction coefficient values
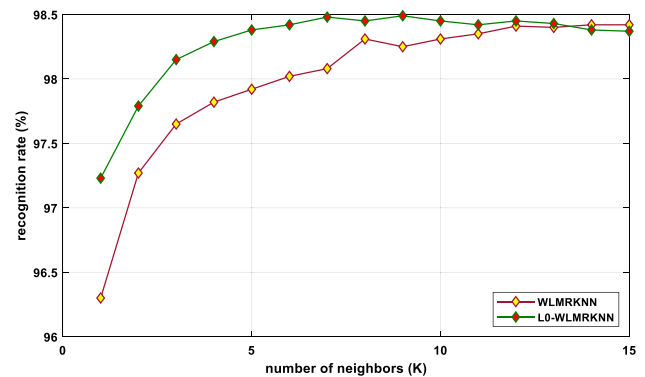
(Eq. 2). On the other hand, increasing the number of neighbors provides more freedom to reconstruct a query sample. It can make the performance of the MLMNN similar to l_0-MLMNN.

The results of the evaluation using SVM classifier

Furthermore, the results of the proposed classifiers are compared with the SVM classifier. The results of the recognition rates are given in Table 7. The results demonstrate the better performance of the proposed classifiers than the SVM classifier on UCI, UCR, and MNIST databases. Also, the results of the proposed K-NN-based classifiers are approximately constant for $K > 5$, which exhibits less sensitivity of them to the initial parameter of the number of neighbors. Generally, the proposed method is a suitable classifier for classifying small datasets.

Discussion of the results

The minimum reconstruction error-based K-NN classifiers have the best performance among K-NN-based classifiers and are less sensitive to the number of neighbors. Usually, the

Table 7 The results of the recognition rates using l_0-WLMRKNN, l_0-WRKNN, l_0-MLMNN, and SVM classifiers

Dataset	Recognition rate (%)			
	l_0-WLMRKNN	l_0-WRKNN	l_0-MLMNN	SVM classifier
Seven UCI datasets	**89.36**	89.21	89.15	86.25
Five UCR datasets	90.52	**92.08**	90.13	80.49
Subset of MNIST	97.28	**97.38**	97.32	89.88

The best results are bolded

neighbors are selected based on the minimum Euclidean distance of the data from the query sample. However, different kinds of K-NN-based classifiers make the decision using their specific criteria. Therefore, neighbor selection according to the criterion of the classifier can improve their performance. The proposed l_0-based neighbor selection method decreases the reconstruction errors of the minimum reconstruction error K-NN-based classifiers and can improve their performance. Figure 16 shows the sum of the reconstruction error values for the query samples according to their category samples on the *Chlorine Concentration* dataset of UCR. The minimum reconstruction error values of query samples (i.e., $r(\boldsymbol{y}, \boldsymbol{X}_K) = \|\boldsymbol{y} - \boldsymbol{X}_K{}^{\mathrm{T}}\boldsymbol{\omega}\|_2^2$) have been calculated using K training samples with the same category of each query sample. The results are provided for $K = 2,\ldots, 15$ using WRKNN and l_0-WRKNN classifiers. The *Chlorine Concentration* dataset includes 3840 query samples, and the number of the categories is 3 ($C = 3$). The results demonstrate the reconstruction errors are decreased using the proposed neighbor selection method.

The reconstruction error of the query sample, i.e., $r(\boldsymbol{y}, \boldsymbol{X}_K) = \|\boldsymbol{y} - \boldsymbol{X}_K{}^{\mathrm{T}}\boldsymbol{\omega}\|_2^2$, is the decision metric in all reconstruction error-based K-NN classifiers. $\boldsymbol{X}_K$ is the matrix of K selected neighbors, and $\boldsymbol{\omega}$ is the reconstruction coefficients vector. $\frac{\partial r}{\partial \boldsymbol{x}_{ik}}$ exhibits the contribution of the $\boldsymbol{x}_{ik}$ in classifying $\boldsymbol{y}$. In l_0-WRKNN, the weighted contribution of each neighbor can be calculated per class as

$$\frac{\partial r^j}{\partial \boldsymbol{x}_{i-l_0}^j} = \frac{\partial \left\| \boldsymbol{y} - \left(\boldsymbol{X}_{K-l_0}^j\right)^{\mathrm{T}} \boldsymbol{\eta}_{K-l_0}^j \right\|_2^2}{\partial \boldsymbol{x}_{i-l_0}}$$
$$= 2\eta_{i-l_0}^j\left(\boldsymbol{y} - \left(\boldsymbol{X}_{K-l_0}^j\right)^{\mathrm{T}} \boldsymbol{\eta}_{K-l_0}^j\right), \qquad (12)$$

where $\boldsymbol{x}_{i-l_0}^j$ is the ith l_0-based neighbor from the jth class, and $\eta_{i-l_0}^j$ is its corresponding coefficient. Equation (12) expresses behavior similar to WRKNN and shows that neighbors have different contributions in classification, corresponding to their reconstruction coefficients. This conclusion is also correct for l_0-WLMRKNN and l_0-MLMNN.
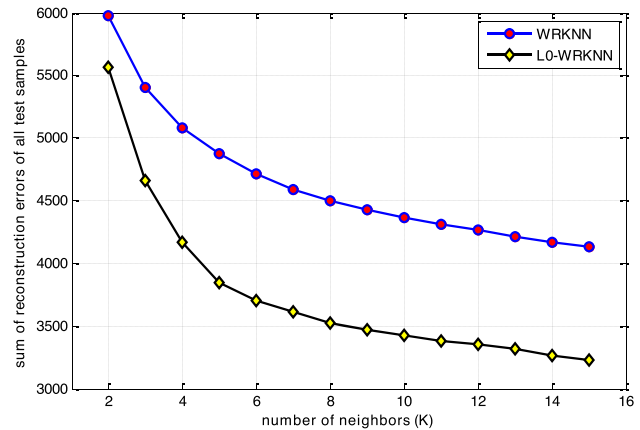

Fig. 16 The sum of the reconstruction error values of all query samples according to their category samples on the *Chlorine Concentration* dataset using WRKNN and l_0-WRKNN classifiers

Besides, by deriving r to the reconstruction coefficients, i.e., $\frac{\partial r}{\partial w_{iK}}$, the weighted contribution of the reconstruction coefficients can be evaluated. In l_0-WRKNN, $\frac{\partial r^j}{\partial \eta_{i-l_0}^j}$ is calculated as

$$\frac{\partial r^j}{\partial \eta_{i-l_0}^j} = \frac{\partial \left\| \boldsymbol{y} - \left(\boldsymbol{X}_{K-l_0}^j\right)^{\mathrm{T}} \boldsymbol{\eta}_{K-l_0}^j \right\|_2^2}{\partial \eta_{i-l_0}^j}$$
$$= 2\left(\boldsymbol{x}_{i-l_0}^j\right)^{\mathrm{T}}\left(\boldsymbol{y} - \left(\boldsymbol{X}_{K-l_0}^j\right)^{\mathrm{T}} \boldsymbol{\eta}_{K-l_0}^j\right). \qquad (13)$$

Equation (13) shows that the weighted contribution of the $\eta_{i-l_0}^j$ depends on both the corresponding sample and the reconstruction error value. Because of selecting neighbors based on the minimum reconstruction error, the proposed l_0-based K-NN classifiers can be less sensitive to the reconstruction coefficients. Generally, Eq. (13) can be applied for evaluating the reconstruction coefficients in every reconstruction error-based K-NN classifier. $f(K, t) = \sum_{j=1}^{C} \sum_{i=1}^{K} \left| \frac{\partial r_t^j}{\partial \eta_i^j} \right|$ is introduced for comparing the sensitivity of the minimum reconstruction error-based K-NN classifiers
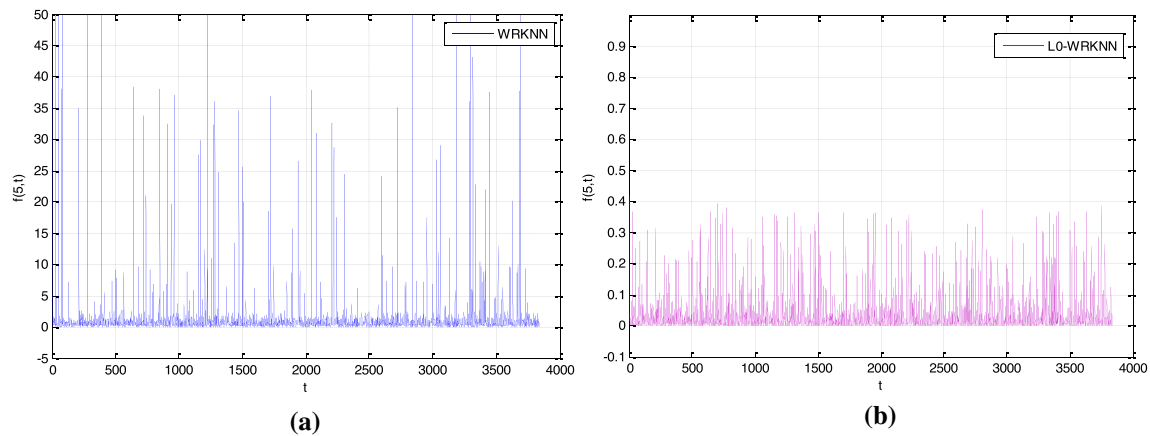
(a)

(b)

Fig. 17 $f(5, .)$ values of the query samples of the *Chlorine Concentration* dataset of UCR using **a** WRKNN and **b** l_0-WRKNN
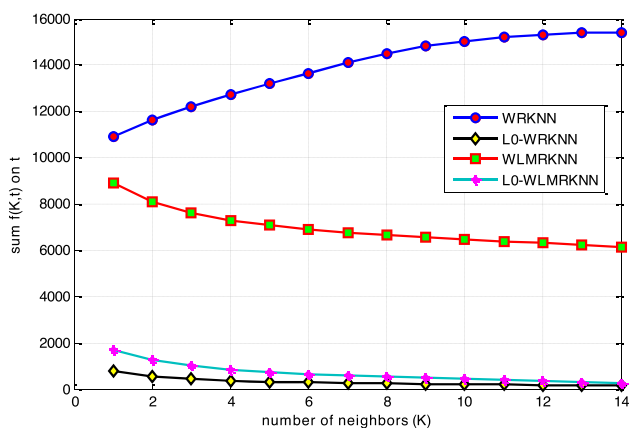

Fig. 18 $\sum_{t=1}^{N_t} f(K, t)$ values for $K = 2,\ldots, 15$ on the *Chlorine Concentration* dataset of UCR using WRKNN, l_0-WRKNN, WLMRKNN, and l_0-WLMRKNN

to the reconstruction coefficients. $\sum_{i=1}^{K} \left| \frac{\partial r_t^j}{\partial \eta_i^j} \right|$ is the summation of K absolute values of $\frac{\partial r_t^j}{\partial \eta_i^j}$ corresponding K neighbors of the tth query sample. And $f(K, t)$ is the summation of c values of $\sum_{i=1}^{K} \left| \frac{\partial r_t^j}{\partial \eta_i^j} \right|$ corresponding to c categories. r_t is the reconstruction error of the tth query sample. The smaller $f(., .)$ exhibits less sensitivity of the reconstruction error-based K-NN classifiers to the reconstruction coefficients.

Figure 17 shows $f(5, .)$ of the query samples of the *Chlorine Concentration* dataset of UCR using WRKNN and l_0-WRKNN. Also, $\sum_{t=1}^{N_t} f(K, t)$ values using WRKNN, l_0-WRKNN, WLMRKNN, and l_0-WLMRKNN are given in Fig. 18 for $K = 2,\ldots, 15$. N_t is the number of query samples. Figures 17 and 18 demonstrate the proposed minimum reconstruction error-based K-NN classifiers are less sensitive to the reconstruction coefficients.

The minimum reconstruction error-based K-NN classifiers make the decision about a query sample based on the minimum reconstruction error. Generally, the reconstruction error of a sample versus the samples with the same class is less than the reconstruction errors versus the samples with a different class. On the other hand, the neighbors corresponding to Euclidean distance cannot always reconstruct the query sample with minimum error. The proposed l_0-based neighbor selection method selects K samples from each class that obtain minimum reconstruction error. Therefore, it is more probable that the samples with the same class as the query sample provide the least reconstruction error.

Conclusion

Deep learning obtains exciting results in many applications, such as classifying includes electroencephalography (EEG) [29], text [13], time-series data [30], remote sensing images [31], etc. However, for training the deep neural network, large datasets are needed. In this manuscript, a robust and powerful K-NN-based classifier is proposed for small datasets. There are different types of K-NN-based classifiers. Neighbors' selection is the first and one of the most significant steps of K-NN-based classifiers. Selecting neighbors according to the decision criterion of the classifier can improve the performance of the classifier. The neighbors are selected using Euclidean distance for most K-NN-based classifiers while are not corresponding to their decision criteria. In this manuscript, an l_0-based neighbor selection method has been introduced (Algorithm 1) for minimum reconstruction error-based K-NN classifiers. There is no constraint on the distance of the selected neighbors, and the neighbors are determined using a sparse representation problem scheme.

Based on the proposed neighbor selection method, l_0-MLMNN, l_0-WRKNN, and l_0-WLMRKNN classifiers have been introduced. Steps of the proposed classifiers have been given in Algorithms 2–4. The reconstruction error of the query sample versus the neighbors has significantly been

decreased using l_0-based neighbors; therefore, the performance of the minimum reconstruction error-based K-NN classifiers has been improved.

Also, the computational complexity of the proposed neighbor selection method ($O(N \times K)$) is more than the conventional minimum Euclidean distance-based method ($O(N)$). Also, the used matching pursuit algorithm is a semi-optimum solution of (11), which can reduce the performance of the proposed method. Furthermore, the proposed neighbor selection method is just applicable for minimum reconstruction error-based classifiers.

The proposed l_0-based neighbor selection method is suitable for data with low samples or high variability per class. The performances of the l_0-based neighbor selection method and conventional Euclidean distance-based method are the same for data with a large number of samples per class or low variability. Evaluations on UCI machine learning repository (Figs. 4, 5 and 6, and Table 3), UCR time-series classification archive (Figs. 7, 8 and 9, and Tables 4, 5), and the subset of the MNIST handwritten digit database (Figs. 10, 11, 12, 13, 14 and 15, and Table 6) demonstrate the suitable performance of the proposed classifiers. It has been shown that the proposed reconstruction error-based K-NN classifiers are less sensitive to the reconstruction coefficients than the conventional minimum reconstruction error-based K-NN classifiers. Also, the proposed classifiers have performed better than the SVM classifier on all three datasets. For future research, the performance of the K-NN-based classifiers can be evaluated using different distance metrics.

Data availability The used datasets are free and available. Also, the codes of the proposed methods are available at: https://github.com/RassoulHajizadeh/Unconstrained-Neighbor-Selection-for-Minimum-Reconstruction-Error-based-K-NN-Classifiers.

References

1. Chen Z, Wu XJ, Cai YH, Kittler J (2021) Sparse non-negative transition subspace learning for image classification. Signal Process 183:107988
2. Yang J, Liu J, Han R, Wu J (2021) Transferable face image privacy protection based on federated learning and ensemble models. Complex Intell Syst 7(5):2299–2315
3. Hajizadeh R, Aghagolzadeh A, Ezoji M (2018) Fusion of LLE and stochastic LEM for Persian handwritten digits recognition. Int J Doc Anal Recognit 21(1):109–122
4. Wang Z, Li X, Duan H, Su Y, Zhang X, Guan X (2021) Medical image fusion based on convolutional neural networks and non-subsampled contourlet transform. Expert Syst Appl 171:114574
5. Goyal B, Lepcha DC, Dogra A, Wang SH (2021) A weighted least squares optimisation strategy for medical image super resolution via multiscale convolutional neural networks for healthcare applications. Complex Intell Syst 8:1–16
6. Wong WK, Juwono FH, Khoo BTT (2021) Multi-features capacitive hand gesture recognition sensor: a machine learning approach. IEEE Sens J 21(6):8441–8450
7. Labintsev A, Khasanshin I, Balashov D, Bocharov M, Bublikov K (2021) Recognition punches in karate using acceleration sensors and convolution neural networks. IEEE Access 9:138106–138119
8. Zhang C, Guo Q, Li Y (2020) Fault detection in the Tennessee Eastman benchmark process using principal component difference based on k-nearest neighbors. IEEE Access 8:49999–50009
9. Jang J, Kim CO (2022) Unstructured borderline self-organizing map: learning highly imbalanced, high-dimensional datasets for fault detection. Expert Syst Appl 188:116028
10. Shlezinger N, Farsad N, Eldar YC, Goldsmith AJ (2021) Model-based machine learning for communications. arXiv preprint. arXiv:2101.04726
11. Raj C, Meel P (2021) ConvNet frameworks for multi-modal fake news detection. Appl Intell 51(11):8132–8148
12. Liu H, Long Z (2020) An improved deep learning model for predicting stock market price time series. Digital Signal Process 102:102741
13. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J (2021) Deep learning-based text classification: a comprehensive review. ACM Comput Surv (CSUR) 54(3):1–40
14. Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27
15. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intell Syst Appl 13(4):18–28
16. Uebele V, Abe S, Lan MS (1995) A neural-network-based fuzzy classifier. IEEE Trans Syst Man Cybern 25(2):353–361
17. Gou J, Qiu W, Yi Z, Shen X, Zhan Y, Ou W (2019) Locality constrained representation-based K-nearest neighbor classification. Knowl Based Syst 167:38–52
18. Li W, Du Q, Zhang F, Hu W (2014) Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery. IEEE Geosci Remote Sens Lett 12(2):389–393
19. Dudani SA (1976) The distance-weighted k-nearest-neighbor rule. IEEE Trans Syst Man Cybern 4:325–327
20. Gou J, Qiu W, Mao Q, Zhan Y, Shen X, Rao Y (2017) A multi-local means based nearest neighbor classifier. In: 2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI). IEEE, Boston, pp 448–452
21. Mitani Y, Hamamoto Y (2006) A local mean-based nonparametric classifier. Pattern Recognit Lett 27(10):1151–1159
22. Zeng Y, Yang Y, Zhao L (2009) Pseudo nearest neighbor rule for pattern classification. Expert Syst Appl 36(2):3587–3595
23. Gou J, Zhan Y, Rao Y, Shen X, Wang X, He W (2014) Improved pseudo nearest neighbor classification. Knowl Based Syst 70:361–375
24. Gou J, Ma H, Ou W, Zeng S, Rao Y, Yang H (2019) A generalized mean distance-based k-nearest neighbor classifier. Expert Syst Appl 115:356–372

25. Gou J, Sun L, Du L, Ma H, Xiong T, Ou W, Zhan Y (2022) A representation coefficient-based k-nearest centroid neighbor classifier. Expert Syst Appl 194:116529

26. Asuncion A, Newman D (2007) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine

27. Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015) The UCR time series classification archive

28. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

29. Craik A, He Y, Contreras-Vidal JL (2019) Deep learning for electroencephalogram (EEG) classification tasks: a review. J Neural Eng 16(3):031001

30. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller PA (2019) Deep learning for time series classification: a review. Data Min Knowl Discov 33(4):917–963

31. Cheng G, Xie X, Han J, Guo L, Xia GS (2020) Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities. IEEE J Sel Top Appl Earth Observ Remote Sens 13:3735–3756