An Efficient Pseudo Nearest Neighbor Classifier

Zheng Chai, Yanying Li*, Aili Wang, Chen Li, Baoshuang Zhang and Huanhuan Gong

***Abstract*—K-nearest neighbor (KNN) rule is a very simple and efficient non-parametric classification algorithm that is widely used in machine learning. In this paper, we proposed a attribute weighting local-mean pseudo nearest neighbor rule (AWLMPNN). The main difference of AWLMPNN and local mean-based pseudo nearest neighbor (LMPNN) is that they use attribute weighting distance and Euclidean distance to measure the distance between two samples, respectively. To illustrate the effectiveness of the proposed AWLMPNN method, extensive experiments on 30 real UCI data sets are conduced by comparing with four competing KNN-based methods. The experimental results show that the proposed AWLMPNN method is superior to other methods, especially in the case of high dimensional attributes with small sample size.**

***Index Terms*—K nearest neighbors, local mean vector, attribute weighting.**

I. Introduction

THE K nearest neighbor (KNN) rule [1] as one of the top ten algorithms in the field of data mining [2]. Since its simple, efficient and competitive advantages, it has been widely used and developed in the fields of pattern recognition and machine learning. To classify a query point, KNN find K nearest neighbors of query pattern from the training data set, and assigns the query point to the majority class among the K nearest neighbor samples.

From the decision-making process of KNN, we can see that KNN is affected by three factors [2], [3]. Firstly, the sensitivity of K value is a key factor in KNN-based classification. In order to select appropriate K values, the cross-validation (CV) method discussed in [4] can obtain different K values according to the size of the training set or the distribution of classes for different data sets, which requires a lot of time. According to the classification accuracy of Leave-One-Out cross validation is approximate concave for the parameter K. [5] designs a search method to select the best value of K. In [6], an adaptive selection of neighborhood size based on statistical confidence is proposed. A dynamic k value selection method for each sample is proposed by using a simple clustering process [7].

Secondly, in the simple majority voting rule adopted by KNN, the selected K nearest neighbors have the same classification contribution to query point, which will lead to enhanced K value sensitivity and weakened classification effect. To solve this problem, some KNN-based weighted voting methods are proposed [8]–[12]. Different weighting functions are used to assign weight to K nearest neighbors of query point. Finally, the KNN algorithm uses Euclidean distance metric to describe the similarity between different samples. This similarity calculation method results in that the KNN algorithm is very sensitive to noise features and susceptible to the influence of non-information features. In order to solve this problem, a simple adaptive distance measure is proposed in [13], Mahalanobis distance is used as the distance measure [14], and both of them can improve the classification effect of KNN. In addition, some representation based KNN methods, such as coarse to fine K nearest neighbor classifier, Collaborative-Representation-Based Nearest Neighbor Classifier for Hyperspectral Imagery, a generalized mean distance-based K-nearest neighbor classifier [15]–[17] can improve the classification performance and generalization ability of KNN classifier to a certain extent.

In addition, there are many improved algorithms based on KNN. For example, some algorithms are proposed in terms of attributes. A nearest neighbor classification based on attribute weighting has been developed in [18], which calculates the weight based on entropy or Gini coefficient. Although it can improve the classification performance, the calculation proposed in that paper is based on the fact that the given attributes are independent of other attributes. In real data sets, attributes are not necessarily independent of each other. A multi-attribute ranking method is proposed [19], and the decision problem and two examples are given to prove the practicability and effectiveness of the method. Lopez et al. [20] proposed redefining nearest neighbor classification in high-dimensional settings. By embedding the filtering method of feature selection into the definition of distance measure, the method encourages only the variables based on the most relevant problems. This method is limited by the number of selected features. Ensemble of a subset of KNN classifiers proposed in [21], feature extraction is combined with KNN algorithm to generate several sub-classifiers. Bootstrap aggregating method is used to integrate to improve classification performance, which requires a lot of time and the number of sub-classifiers to be generated according to different data sets. In order to reduce the negative impact of existing outliers, especially in the case of small samples,

Manuscript received March 6, 2021; revised August 31, 2021. The research was supported by the National Natural Science Foundation of China (Grant No. 61573266, 11801013) and the Foundation of the Doctoral Scientific Research of Baoji University of Arts and Sciences(Grant No.209040122)and the Foundation of Baoji City Guidance plan for scientific and technological innovation(Grant No.2017JH2-23, 2018JH-19). Scientific Research Program Funded by Shaanxi Provincial Education Department (Grant No. 18JK0045). The research was supported by Natural Science Major Research Plan in Shaanxi Province of China(Grant No.2018ZKC-115, 2018ZKC-023).

Zheng Chai is Postgraduate of the School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721013, Shaanxi, China. (e-mail: 15735159302@163.com).

Yanying Li is Associate Professor of the School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721013, Shaanxi, China. (Corresponding author, e-mail: liyanying2021@163.com).

Aili Wang is Professor of the School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721013, Shaanxi, China. (e-mail: aily_wang83@163.com).

Cheng Li is Postgraduate of the School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721013, Shaanxi, China. (e-mail: 1248472750@qq.com).

Baoshuang Zhang is Postgraduate of the School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721013, Shaanxi, China. (e-mail: 969800643@qq.com).

Huanhuan Gong is Postgraduate of the School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721013, Shaanxi, China. (e-mail: 2715090881@qq.com).

Mitani and Hamamoto proposed local mean-based K-nearest neighbor (LMKNN) rule [22]. LMKNN compute the distance between query point and the local mean vector of every class, and classifies the query point to the class with the smallest distance among all classes. In pseudo nearest neighbor (PNN) [23] the pseudo nearest neighbor is determined by the weighted sum of the distance between the query point and the k nearest neighbors of every class. Then, the class label of the class of the pseudo nearest neighbor nearest to the query sample is assigned to the query sample. Based on the idea that LMKNN constructing local mean vector, some KNN-based algorithms have emerged further overcome the existing outliers and improve the classification performance. Local mean-based pseudo nearest neighbor (LMPNN) [24], LMPNN firstly calculates K local mean vectors by K nearest neighbors of the query point in every class. Then, the distance between the query point and the pseudo nearest neighbor of every class is defined as the sum of the weighted distances between the query point and its K local mean vectors. lastly, query point are classified into the class of the pseudo nearest neighbor with the smallest distance to query point. All the above methods have been proved to be robust to outliers, with low sensitivity to the K value, and satisfactory classification results.

To further improve the performance of the KNN-based classifier, a new attribute weighting method was proposed, and it is combines with LMPNN to generate AWLMPNN algorithm. Different from LMPNN, AWLMPNN uses attribute weighting distance instead of Euclidean distance to measure the distance between two samples. Compared with Euclidean distance, the attribute weighting distance can fully consider the contribution of different attributes to classification, so as to determine a more appropriate nearest neighbor and achieve better classification performance. To verify the classification effect of the proposed AWLMPNN method, we conducted experiments with other four KNN-based methods on 30 real data sets. A large number of experiments show that the proposed AWLMPNN method has good robustness and classification performance.

The rest of this paper is organized as follows. In the Section 2, we briefly review the related KNN-based methods. In the Section 3, AWLMPNN method is proposed and analyzed in detail. In the Section 4, we summarize the comparative experiments on several real data sets. Finally, we give the conclusion in the Section 5.

II. The related classification

In this part, we will briefly introduce KNN algorithm. Let $T = \{x_i \in R^d\}_{i=1}^N$ be a training set with N training samples in d-dimensional feature space, and there are m class labels, each sample $x_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ corresponds to its class label y_i, where $y_i \in \{c_1, c_2, \cdots, c_m\}$. let $T^j = \{x_i^j \in R^d\}_{i=1}^{N_j}$ denote a class of T from the class c_j, with the number of the training samples N_j. For a query point x, the KNN rule is implemented as follows:

1) Computing the distances between the query point x and samples from the training sample set T. Then find K nearest neighbors set $T_K(x) = \{x_i \in R^d\}_{i=1}^K$ of x according to these distances. The distance between x and the neighbor x_i is measured by the Euclidean distance metric by Eq(1).

$$d(x, x_i) = \sqrt{(x - x_i)^T (x - x_i)}. \tag{1}$$

2) The query point x are classified as the most frequent class in $T_K(x) = \{x_i \in R^d\}_{i=1}^K$.

$$c = \arg\max_j \sum_{x_i \in T_K(x)} sign(x_i, c_j). \tag{2}$$

where x_i is one of the neighbors in the training set, $sign(x_i, c_j) \in \{0, 1\}$ is a indicator function that implies x_i whether belongs to class c_j.

III. The proposed AWLMPNN method

A. motivation

In machine learning, KNN-based classification methods have become the most attractive classification algorithm in many practical applications because of its simple, effective, intuitive and competitive advantages. However, as discussed in Section I, their performance is still affected by the existing distance measures, especially in the case of high dimensional attributes with small sample size. When making a classification decision, the class of the query point depends on its K nearest neighbors. So finding the right nearest neighbor for the query point is crucial. However, in the traditional Euclidean metric, Chebyshev metric, Manhattan metric and so on, different attributes have the same contribution to the determination of the nearest neighbor of query point. But in practical problems, attributes often contribute differently to the classification. Therefore, in order to make full use of the contribution of different attributes to find a more suitable nearest neighbor, this paper designs an attribute weighting distance. The distance of two d dimension attribute samples x_s, x_t is expressed as $d_m(x_s, x_t)$:

$$d_m(x_s, x_t) = \sum_{l=1}^{d} w_l \times |x_{sl} - x_{tl}|. \tag{3}$$

The attribute weight w_l $(l = 1, 2, \cdots, d)$ is learn as follows:

1) Generate d single attribute sample sets $T_l = \{x_{il} \in R\}_{i=1}^N, (l = 1, 2, \cdots, d)$ from the training sample set T by feature extraction, where the class label of $x_{il}, (l = 1, 2, \cdots, d)$ is y_i, $y_i \in \{c_1, c_2, \cdots, c_m\}$.

2) In the KNN-based classifier which uses Manhattan distance as the distance measure, $T_l (l = 1, 2, \cdots, d)$ is verified by Leave-One-Out Cross Validation, then the classification accuracy is recorded as a_l.

3) According to the classification accuracy a_l, the weight of attributes is given, and the weight of the l-th attribute is w_l:

$$w_l = \frac{a_l}{\sum_{l=1}^{d} a_l} \qquad l = 1, 2, \cdots, d. \tag{4}$$

Compared with the traditional Euclidean distance, the attribute weighting distance fully considers the different contribution of different attributes to the classification effect, through determine a more appropriate nearest neighbor for the query point to improving the classification performance.

B. the AWLMPNN classification rule

The proposed AWLMPNN method determines the label of query sample x through the following procedures:

1) The K nearest neighbors of x are found from T^j of each class c_j in the training set T and represented by $T_K^j(x) = \{x_i^j \in R^d\}_{i=1}^K$. The K nearest neighbors $x_1^j, x_2^j, \cdots, x_K^j$ are sorted in ascending order in the light of their attribute weighting distance to x. The attribute weighting distance $d_m(x, x_i^j)$ between query point x and x_i^j is as follows:

$$d_m(x, x_i^j) = \sum_{l=1}^{d} w_l \times |x_l - x_{il}^j|. \tag{5}$$

2) Calculate the local mean vector $\overline{x_i}^j$ of the first i nearest neighbors of the query point x from class c_j

$$\overline{x_i}^j = \frac{1}{i}\sum_{l=1}^{i} x_l^j \qquad i = 1, 2, \cdots, K. \tag{6}$$

let $\overline{T_K}^j = \{\overline{x_i}^j \in R^d\}_{i=1}^K$ donate the set of the K local mean vectors corresponding to the K nearest neighbors in the class c_j, $d_m(x, \overline{x_i}^j), i = 1, 2, \cdots, K.$ are their attribute weighting distances to x.

3) Different weights are given to K local mean vectors, the weight $\overline{w_i}^j$ of the i-th local mean vector $\overline{x_i}^j$ from class c_j is calculate as follows:

$$\overline{w_i}^j = \frac{1}{i} \qquad i = 1, 2, \cdots, K. \tag{7}$$

4) Find local mean-based pseudo nearest neighbor of the query point x from per class. let $\overline{x_j}^{PNN}$ denote local mean-based pseudo nearest neighbor of x from class w_j. The distance $d_m(x, \overline{x_j}^{PNN})$ between x and $\overline{x_j}^{PNN}$ is calculated as:

$$\begin{aligned} d_m(x, \overline{x_j}^{PNN}) = (\overline{w_1}^j \times d_m(x, \overline{x_1}^j) + \overline{w_2}^j \times \\ d_m(x, \overline{x_2}^j) + \cdots + \overline{w_K}^j \times d_m(x, \overline{x_K}^j)). \end{aligned} \tag{8}$$

5) According to Eq(8), x is classified into class that has the closest local mean-based pseudo nearest neighbor among all classes.

$$c = \arg\min_{c_j} d_m(x, \overline{x_j}^{PNN}). \tag{9}$$

C. the proposed algorithm

AS mentioned in Section 3.2 the pseudo codes of AWLMPNN algorithm is shown in Algorithm 1.

Algorithm 1:The proposed AMLMPNN method.

Input:
x: a query point ,$T = \{x_i \in R^d\}_{i=1}^N$: training set,
$T^j = \{x_i^j \in R^d\}_{i=1}^{N_j}$: training subset from class c_j,
K: the neighborhood size, m: the number of classes in T.
$c_1, c_2, \cdots, c_m$: m class labels, $N_1, N_2, \cdots, N_m$: the numbers of training samples of m classes.

Output:
c: Class label of query point x.

Step 1: Generate d single attribute sample sets $T_l = \{x_{il} \in R\}_{i=1}^N$ $(l = 1, 2, \cdots, d)$ from the training sample set T by feature extraction.

Step 2: for l=1 to d **do**
In the KNN-based classifier which uses Manhattan distance as the distance measure, $T_l (l = 1, 2, \cdots, d)$ is verified by Leave-One-Out Cross Validation, and the classification accuracy is recorded as a_l.
end for

Step 3: Calculate the weight of d attributes.
for l=1 to d **do**
$w_l = \frac{a_l}{\sum_{l=1}^{d} a_l}$
end for
Then set $W = \{w_1, w_2, \cdots, w_d\}$

Step 4: Calculate the distance between x and the sample in per class c_j. **for** i=1 to N_j **do**
$d_m(x, x_i^j) = \sum_{l=1}^{d} w_l \times |x_l - x_{il}^j|$
end for

Step 5: Find the K nearest neighbors of x from T^j. The K nearest neighbors are sorted in ascending order according $d_m(x, x_i^j)$, say $T_K^j(x) = \{x_i^j \in R^d\}_{i=1}^K$.

Step 6: Use $T_K^j(x)$ to compute the local mean vector $\overline{x_i}^j$ of the first i nearest neighbors of x , then calculate the distance $d_m(x, \overline{x_i}^j)$ between x and $\overline{x_i}^j$.
for i=1 to K **do**
$\overline{x_i}^j = \frac{1}{i}\sum_{l=1}^{i} x_l^j$
$d_m(x, \overline{x_i}^j) = \sum_{l=1}^{d} w_l \times |x_l - x_{il}^j|$
end for
Set $\overline{T_K}^j(x) = \{\overline{x_i}^j \in R^d\}_{i=1}^K$ and
$\overline{D_K}^j = \{d_m(x, \overline{x_1}^j), d_m(x, \overline{x_2}^j), \cdots, d_m(x, \overline{x_K}^j)\}$.

Step 7: Give the weight $\overline{W_i}^j$ to the i-th local mean vector in the set $\overline{T_K}^j(x)$.
for i=1 to K **do**
$\overline{W_i}^j = \frac{1}{i}$
end for
Set $\overline{W_i}^j = \{\overline{W_1}^j, \overline{W_2}^j, \cdots, \overline{W_K}^j\}$.

Step 8: Use $\overline{W_i}^j$ and $\overline{D_K}^j$ to find the pseudo nearest neighbor based on the local mean vector $\overline{x_j}^{PNN}$ according to the following formula.
$d_m(x, \overline{x_j}^{PNN}) = (\overline{w_1}^j \times d_m(x, \overline{x_1}^j) + \overline{w_2}^j \times d_m(x, \overline{x_2}^j).$
$+ \cdots + \overline{w_K}^j \times d_m(x, \overline{x_K}^j))$

Step 9: Classify the query point x to c.
$c = \arg\min_{c_j} d_m(x, \overline{x_j}^{PNN})$

IV. Experiments

To verify the classification performance of proposed AWLMPNN, we compare AWLMPNN with KNN, LMKNN, PNN and LMPNN. Classification accuracy and F1 score are commonly used in classification performance evaluation [25], [26], so we will compare the classification accuracy and F1 score on 30 real data sets from UCI [27].

In addition, we use Wilconxon Signed-Ranks test, Friedman test and T-test to illustrate the advantages of the proposed method. Since in machine learning, nonparametric statistical testing plays an important role in comparing the performance of classifiers on multiple data sets [28], [29].

A. Data information

In this section, we will briefly describe the information of all data sets encountered in the experiment. 30 real data sets were collected from UCI machine learning database. All data information including sample size, number of attributes and number of classes are shown in table I. In order to use the attribute weighting distance, all data sets need to be preprocessed. First, some attribute values are digitized. Secondly, in order to meet the experimental needs, some data sets need to remove the small class samples. Finally, We need to normalize the column values of the training set. For example, we need to remove some classes with a very small size of samples from the data of 'Dermatology', 'Ecoli', 'Housevote', 'Lymphography' and 'Cleveland'. In these data sets, the maximum number of samples and the

TABLE I: The UCI data sets.

Data	Samples	Attributes	Classes
Audi	772	17	2
Balance	625	4	3
Bands	365	19	2
Breast	683	9	2
Coimbra	116	10	2
Cryotherapy	90	6	2
Duser	403	6	4
Dermatology	338	34	5
Diabetes	520	16	2
Ecoli	220	7	2
Hayesroth	160	4	3
Heart	299	13	2
Hepatitis	80	19	2
HillValley	606	100	2
Housevote	232	16	2
Ionosphere	351	34	2
Lymphography	142	18	2
Newthyroid	215	5	3
Parkinsons	195	22	2
Pima	532	7	2
Cleveland	284	13	4
QSAR	1055	41	2
Seed	210	7	3
Robot	5456	4	4
Sonar	208	60	2
Tea	151	5	3
Vowel	990	13	11
Wdbc	569	30	2
Wine	178	3	3
Wpbc	198	32	2

minimum number of samples are 5456 and 80 respectively, and the maximum number of attributes and the minimum are 100 and 4 respectively.

B. Experiments on real data sets

To evaluate the effectiveness and robustness of the proposed AWLMPNN algorithm, experimental contrasts with KNN, LMKNN, PNN and LMPNN are conduct on 30 UCI data sets. These experiments were carried out by 10-fold cross validation. The training samples are randomly selected from each real data set, and the rest of the samples constitute the test set. In the experiment, the K value traverses from 1 to 20, and the step size is 1. We have done 10 experiments on every data set. The ultimate classification performance of every algorithm for per data set is determined by the average of 10 classification accuracies and F1 scores. F1 score is given by

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}. \tag{10}$$

On the basis of F1 score, Micro-F1 (F1 score regardless of classes) and Macro-F1 (mean of F1 scores within the class) were applied for estimating.

Experimental results of all used algorithms on the selected 30 datasets are shown in Table II, III and IV. The best classification performance for each data set is shown in bold. Table II shows the best classification accuracies of each method on each real data set, and the corresponding K value and standard deviation. Tables III and IV show the results in terms of Macro-F1 and Micro-F1, respectively. According to the tables, it is obvious that the proposed AWLMPNN method achieves the best results in these real UCI data sets compared with the other four competitive methods. Two important facts need to be pointed out. First of all, in the case of high dimensional attributes with small sample size, such as 'Sonar', 'Bands', 'Coimbra', 'Heart', 'Parkinsons', 'wdbc', 'Diabetes' and other data, the performance of AWLMPNN is always significantly better than PNN, LMKNN, KNN and LMPNN. This obviously means that the proposed AWLMPNN method is superior to other methods. Secondly, among all the classification tasks, the proposed AWLMPNN achieves the best classification accuracies in 23 out of 30 data sets, while KNN, LMKNN, PNN and LMPNN achieve the best classification accuracies on 2, 2, 3 and 2 data sets, respectively. Moreover, the F1 experimental results in Table III and IV show that compared with the other algorithms, the proposed AWLMPNN has a large advantage in F1 score. The reason for this phenomenon is that compared with PNN, LMKNN, KNN and LMPNN, AWLMPNN can get appropriate nearest neighbors to make correct classification decision. Finally, on all real data sets, AWLMPNN obtains the highest average classification accuracy and the smallest average standard deviation, which indicates that proposed AWLMPNN has the most stable classification performance.

To further prove the robustness of AWLMPNN, the classification performance of AWLMPNN on each real data sets with different K values is studied, and compared with PNN, LMKNN, LMPNN and KNN. The experimental results are shown in Figs. 1, 2, 3 and 4. It is worth noting that in the K-value range, the classification effect of AWLMPNN is better than the other four methods, and the classification difference between AWLMPNN and PNN, LMKNN, LMPNN, KNN is very significant in most cases, especially when the K value is large. As shown in Figs. 1, 2, 3 and 4, we can observe that the classification accuracy of PNN, LMKNN, KNN and LMPNN mostly fluctuates with the change of K value, while the classification accuracy of AWLMPNN is relatively less affected. It is obvious from Figs. 1 and 2 that within the interval of K, AWLMPNN is significantly better than the other four methods. Therefore, the results of Figs. 1, 2, 3 and 4 show that, compared with the competition method, the proposed AWLMPNN is the most robust to the neighborhood size K, and has good classification performance.

However, the comparison of competing classifiers in classification accuracy is not statistically credible. Here, we further use three nonparametric statistical tests: Wilconxon Signed-Ranks test [30], Friedman test and T-test [31] to compare the performance of the classifier.

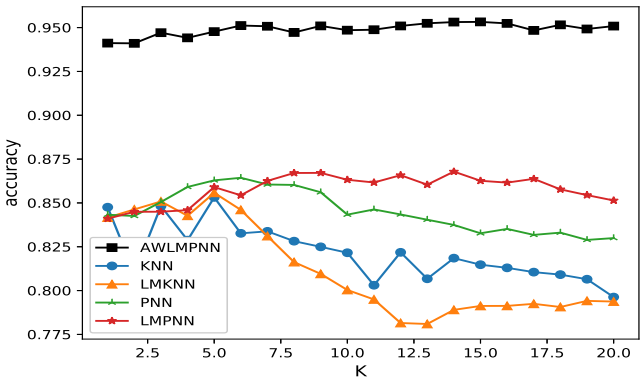
Firstly, according to the results in table II, Wilconxon Signed-Ranks test is used to compare AWLMPNN with KNN, LMKNN, PNN and LMPNN in pairs. Let β_i be the divergence between the accuracy in classification of two algorithms on the i-th among n data sets. Then β_i are sorted in the light of their absolute values, the corresponding ranking of β_i are recorded as $rank(\beta_i)$. Let R^+ denote the rankings sum of the data set whose one algorithms is superior to the other, R^- be the opposite. It should be noted that when $\beta_i = 0$, the ranking values of $\beta_i = 0$ should be divided equally into R^+ and R^-. If the number of $\beta_i = 0$ is odd, a ranking value will be ignored. R^+ and R^- are compute as follow:

TABLE II: The highest accuracy rates(%) of every algorithm with the relevant stds(standard deviations) and K value in the parentheses.

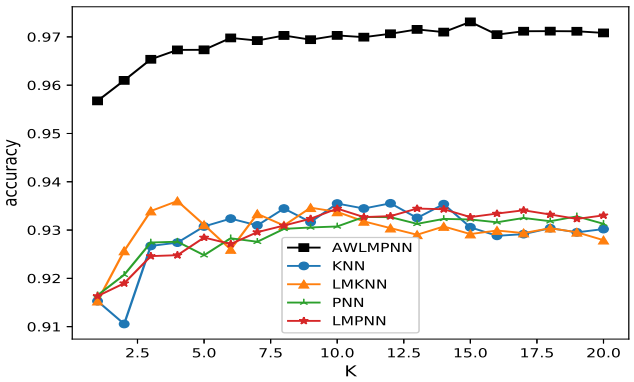
Data	KNN	LMKNN	PNN	LMPNN	AWLMPNN
Audi	97.55±3.81(1)	97.3±0.94(1)	97.38±1.08(1)	97.64±0.1(13)	**99.83 ±0.15(1)**
Balance	89.81±3.67(20)	**92.02±3.95(13)**	90.07±3.8(11)	90.97±3.94(19)	90.19±3.35(19)
Bands	71.02±2.9(1)	70.99±3.53(1)	71.76±1.95(2)	71.81±1.09(2)	**76.58±0.4(11)**
Breast	97.29±0.56(5)	**97.51±0.47(7)**	97.35±0.38(8)	97.19±0.37(17)	97.23±0.22(14)
Coimbra	71.2±1.94(7)	58.61±3.05(1)	60.84±2.52(1)	62.28±1.50(16)	**73.49±2.03(20)**
Cryotherapy	91.89±8.75(1)	92.33±8.66(1)	91.78±3.8(1)	92±2.12(2)	**93.33±1.86(14)**
Duser	87.74±1.51(5)	92.73±1.89(7)	90.67±1.32(10)	91.43±1.8(18)	**93.5±1.04(19)**
Dermatology	89.71±4.82(3)	93.35±3.18(3)	90.98±2.2(2)	95.14± 1.25(14)	**97.63±0.32(9)**
Diabetes	93.29±4.86(1)	94.5±1.96(3)	94.1±0.4(5)	95.37±0.67(20)	**97.83±0.62(1)**
Ecoli	98.14±0.82(3)	98.86±0.46(4)	98.68±0.46(3)	**99.09±0.6(8)**	98.95±0.45(13)
Hayesroth	71.33±7.32(1)	77.12±4.93(3)	71.34±1.46(7)	**77.41±2.13(10)**	75.65±0.90(8)
Heart	72.27±1.43(15)	60.4±1.82(12)	67.03±3.5(19)	62.37±1.88(18)	**75.44±1.32(20)**
Hepatitis	64.5±2.78(17)	64.75±2.4(4)	62.12±1.98(13)	66.38±3.82(17)	**72.5±1.57(20)**
HillValley	57.47±2.21(1)	59.75±2.34(6)	57.86±1.34(1)	60.78±0.85(13)	**61.54±0.98(8)**
Housevote	92.59±0.56(5)	94.93±0.87(19)	92.97±0.33(10)	94.62±0.7(18)	**94.94±0.52(2)**
Ionosphere	89.72±1.48(2)	90.13±0.86(13)	86.5±0.77(3)	89.49±0.71(19)	**90.97±0.2(1)**
Lymphography	79.45±1.07(7)	84.78±1.09(4)	82.86±1.15(4)	84.63±1.02(16)	**88.45±2.29(19)**
Newthyroid	94.54±2.6(1)	95.9±1.23(3)	94.37±1.53(1)	95.71±0.61(5)	**97.43±0.37(4)**
Parkinsons	85.32±1.54(5)	85.58±2.57(5)	86.43±1.15(6)	86.79±0.81(14)	**95.33±0.35(15)**
Pima	75.68±1.97(10)	**78.29±2.97(16)**	73.9±1.16(16)	74.32±1.46(19)	75.9±1.73(19)
Cleveland	55.87±3(18)	55.15±3.73(20)	55.57±3.49(17)	50.74±1.87(20)	**60.85±0.89(11)**
QSAR	82.19±1.36(7)	83.59±1(20)	82.65±0.76(14)	83.1±0.85(20)	**86.9±0.9(11)**
Seed	**94.48±0.94(4)**	90.86±0.49(14)	90.67±0.64(13)	91.19±0.62(5)	93.33±0.31(18)
Robot	97.27±1(1)	97.29±0.71(1)	97.33±0.39(1)	97.45±0.08(2)	**97.73±0.13(2)**
Sonar	82.34±5.73(1)	83.94±2.65(4)	83.5±0.38(14)	85.79±0.87(19)	**89.57±0.85(7)**
Tea	60.53±4.94(1)	57.85±4.75(1)	56.74±1.95(1)	59.53±1.92(2)	**62.37±1.12(17)**
Vowel	99.2±24(1)	99.22±12.5(1)	**99.31±1.44(1)**	99.27±0.3(1)	99.25±0.18(6)
Wdbc	93.55±0.62(12)	93.6±0.43(4)	93.28±0.42(19)	93.45±0.51(13)	**97.31±0.38(15)**
Wine	75.34±1.72(1)	76.32±1.15(1)	76.41±1.90 (3)	79.15±0.96(15)	**98.55±0.60(15)**
Wpbc	76.17±2.64(20)	72.8±2.17(20)	75.85±3.22(19)	71.4±1.65(20)	**79.07±1.43(20)**
Average	82.92±3.42	83.01±2.65	82.34±1.56	83.22±1.24	**87.06±0.92**

TABLE III: The highest Macro-F1(%) of every algorithm with the relevant stds(standard deviations) and K value in the parentheses.

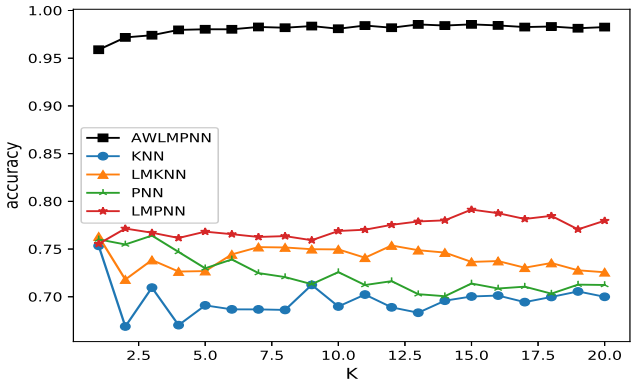
Data	KNN	LMKNN	PNN	LMPNN	AWLMPNN
Audi	97.18±3.8(1)	97.18±10.2(1)	97.2±1.11(1)	97.52±0.13(13)	**99.82 ±0.18(3)**
Balance	62.6±1.97(10)	**77.46±4.95(3)**	62.95±2.21(10)	69.82±4.1(18)	68.24±3.62(20)
Bands	66.43±4.8(1)	66.53±3.95(1)	67.12±4.52(2)	67.72±1.53(2)	**73.85±0.24(11)**
Breast	97.06±0.71(5)	**97.21±0.49(9)**	97±0.44(8)	96.92±0.41(18)	96.95±0.24(14)
Coimbra	56.39±3.38(1)	57.56±3.28(1)	55.84±2.87(1)	59.54±1.84(18)	**71.48±1.74(20)**
Cryotherapy	90.89±8.8(1)	91.54±9.39(1)	90.80±4.45(2)	91.22±2.47(2)	**92.64±1.74(11)**
Duser	86.48±1.9(6)	91.58±2.06(7)	89.34±1.58(13)	90.27±1.87(20)	**92.62±1.14(13)**
Dermatology	86.20±6.45(1)	91.47±3.82(6)	88.41±2.99(4)	92.97±1.71(16)	**96.94±0.45(14)**
Diabetes	93.07±4.9(1)	94.32±2.15(4)	93.86±0.54(5)	95.01±0.73(19)	**97.81±0.67(2)**
Ecoli	97.75±0.9(3)	98.74±0.56(14)	98.47±0.56(4)	**98.99±0.73(13)**	98.83±0.48(20)
Hayesroth	69.22±7.55(1)	76.02±4.99(3)	70.65±2.83(9)	**77.82±3.28(8)**	75.92±1.51(5)
Heart	50.42±3.06(5)	51.2±1.36(5)	49.2±1.92(4)	49.87±1.23(3)	**66.11±1.17(11)**
Hepatitis	58.76±3.63(17)	61.15±2.96(13)	54.52±1.3(12)	63.48±4.3(17)	**65.86±1.24(18)**
HillValley	58.04±2.24(1)	59.51±2.32(7)	57.7±1.6(1)	60.02±0.7(11)	**61.19±0.86(9)**
Housevote	92.39±0.54(9)	**94.82±0.92(7)**	92.62±0.44(8)	94.56±0.75(17)	94.51±0.54(2)
Ionosphere	87.34±1.99(2)	88.49±1.12(13)	83.89±1.13(1)	87.13±0.71(19)	**89.3±0.22(17)**
Lymphography	78.4±1.68(7)	83.37±1.5(6)	80.24±1(4)	82.94±1.29(10)	**86.84±2.25(17)**
Newthyroid	90.26±5.17(1)	92.81±2.09(3)	91.12±2.72(1)	92.76±1.16(8)	**95.3±0.78(5)**
Parkinsons	78.16±5.37(1)	79.29±6.51(1)	80.21±4.23(8)	79.77±1.19(7)	**93.48±0.57(12)**
Pima	72.24±1.8(4)	**74.51±3.32(17)**	68.92±1.15(18)	70.34±1.69(20)	72.17±1.96(19)
Cleveland	26.21±1.32(8)	30.31±1.24(13)	25.35±0.93(1)	26.83±0.55(7)	**39.54±0.83(5)**
QSAR	80.18±1.14(3)	80.95±0.85(8)	80.84±0.78(15)	80.85±0.77(20)	**85.09±0.91(17)**
Seed	90.52±0.77(10)	90.59±0.68(2)	90.21±0.69(1)	90.49±0.59(3)	**92.88±0.32(16)**
Robot	96.88±1.01(1)	96.95±0.77(1)	96.94±0.45(1)	97.02±0.08(2)	**97.30±0.18(2)**
Sonar	81.62±6.12(1)	82.52±2.52(4)	82.81±0.45(12)	84.66±0.76(19)	**88.92±0.93(7)**
Tea	56.92±4.62(1)	55.03±5.2(1)	55.97±2.03(1)	56.44±1.63(1)	**60.45±1.45(1)**
Vowel	**99.35±24.2(1)**	99.28±12.9(1)	99.31±1.5(1)	99.26±0.28(1)	99.22±0.17(6)
Wdbc	92.83±6.3(14)	92.79±0.42(3)	92.74±0.52(11)	93.06±0.62(20)	**97.06±0.45(16)**
Wine	73.78±2.57(1)	73.9±1.26(1)	74.35±2(3)	76.51±1.17(17)	**98.45±0.73(11)**
Wpbc	51.62±2.94(3)	**55.93±3.46(2)**	52.92±2.25(2)	53.08±1.51(5)	51.62±0.69(3)
Average	77.31±4.05	79.43±2.93	77.38±1.71	79.23±1.33	**83.35±0.95**



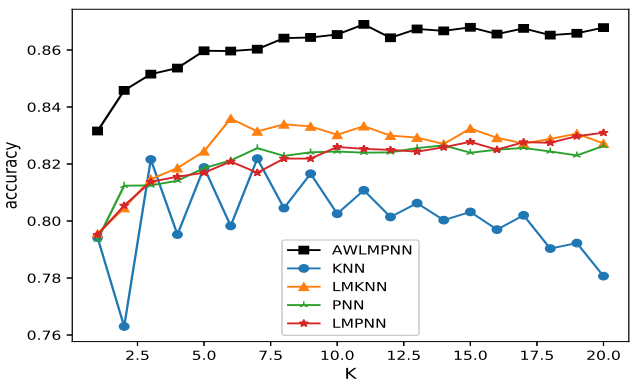
1 Parkinsons



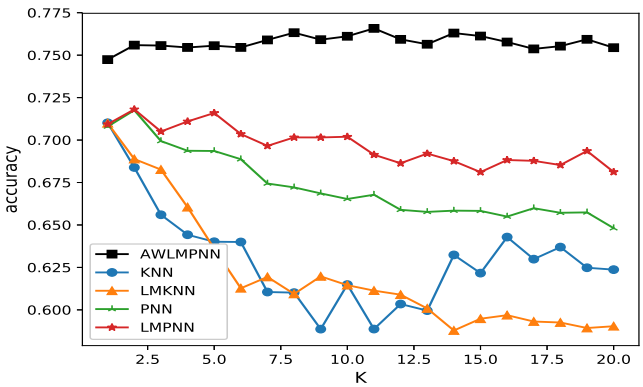
2 Wdbc



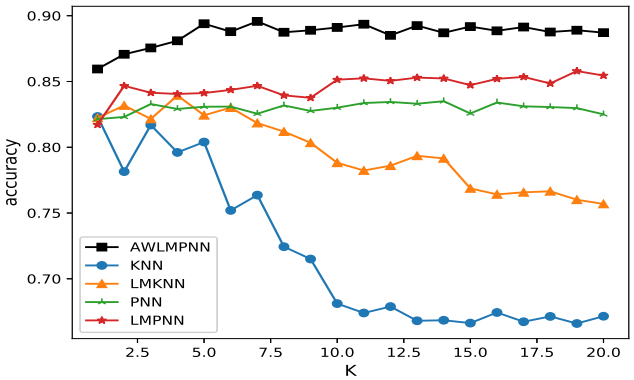
3 Wine



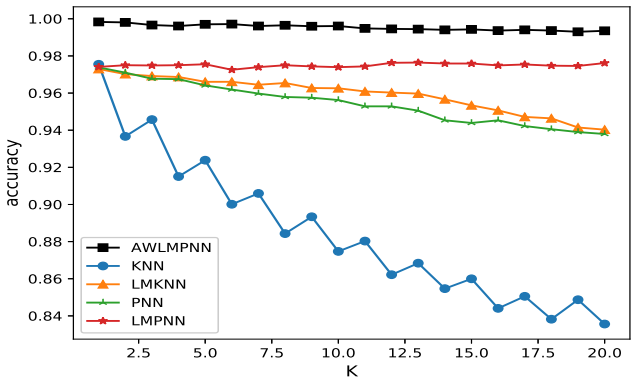
4 QSAR



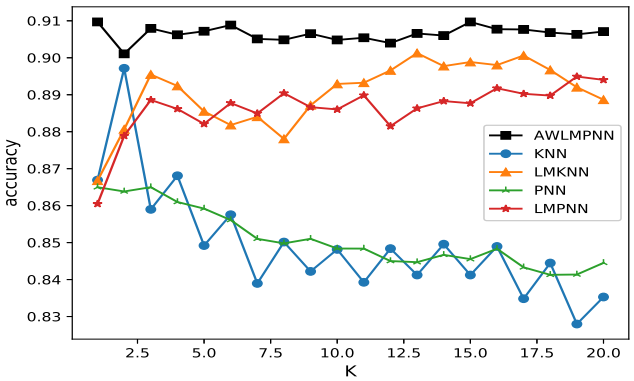
5 Bands



6 Sonar

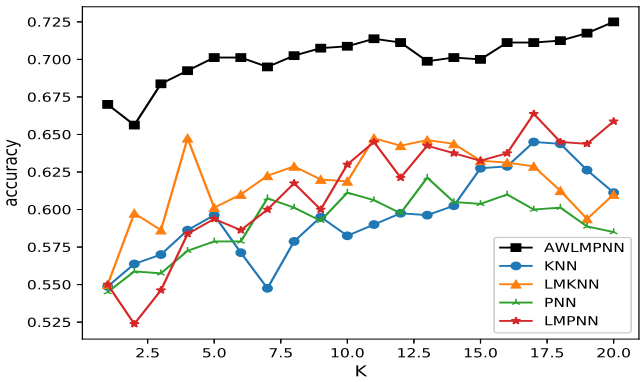


7 Audi

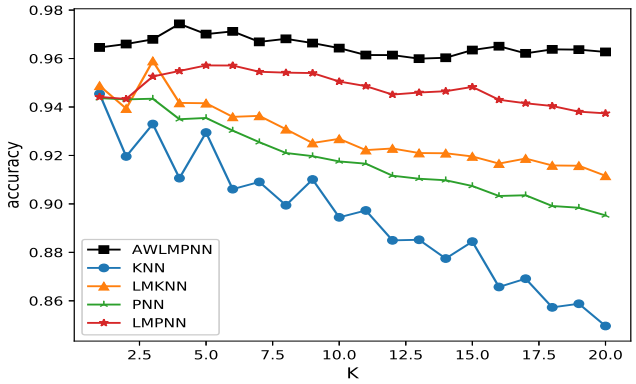


8 Ionosphere

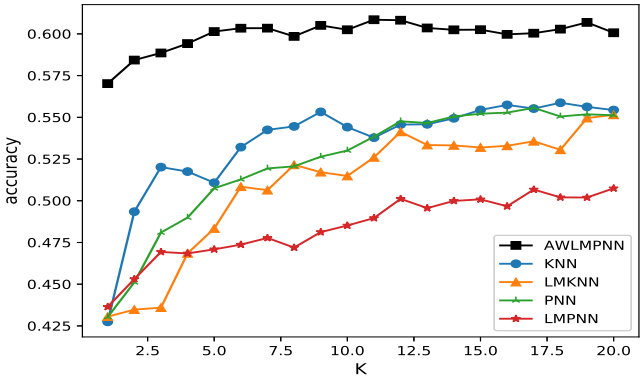
Fig. 1: The classification accuracies of every algorithm via K on all data sets.



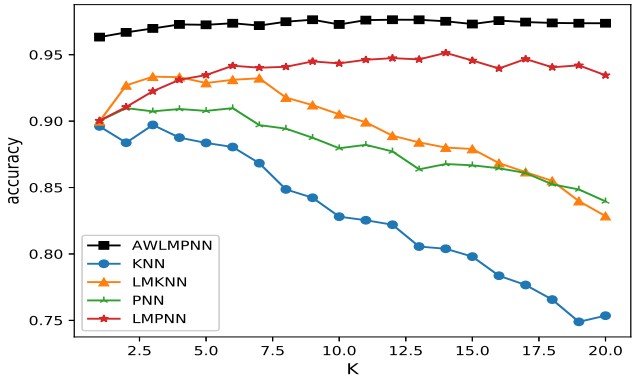
1 Hepatitis



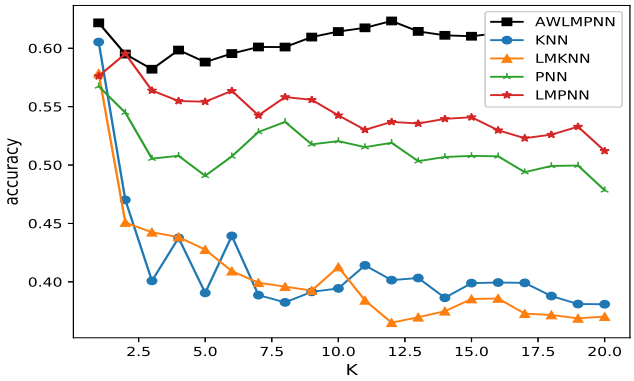
2 Newthyroid



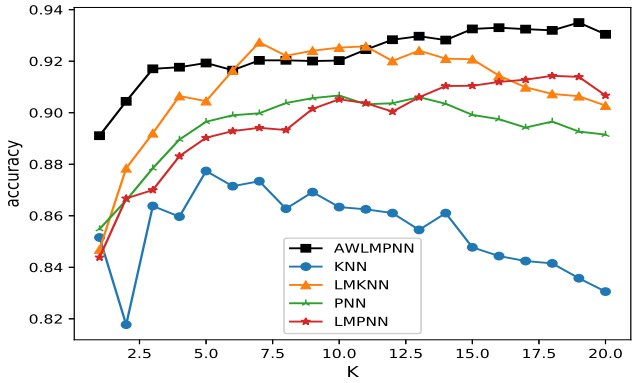
3 Cleveland



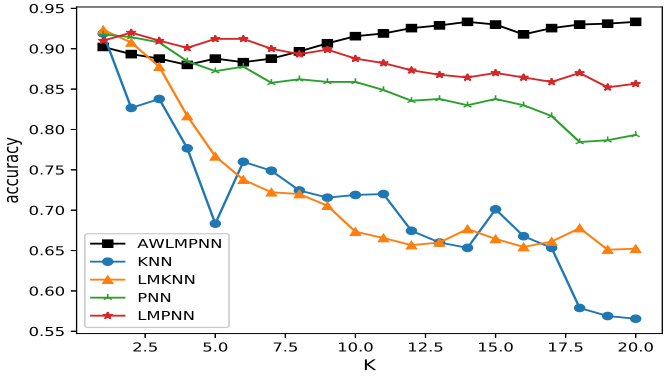
4 Dermatology



5 Tea

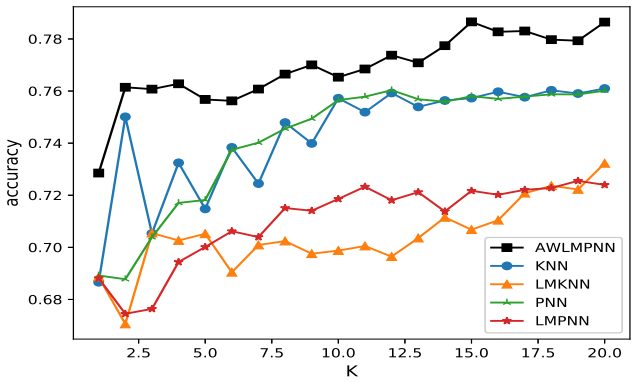


6 Duser

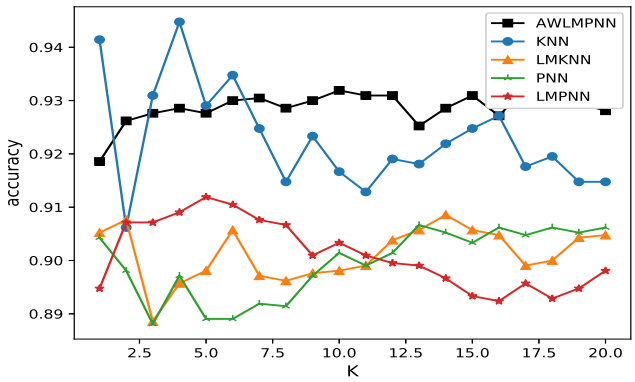


7 Cryotherapy

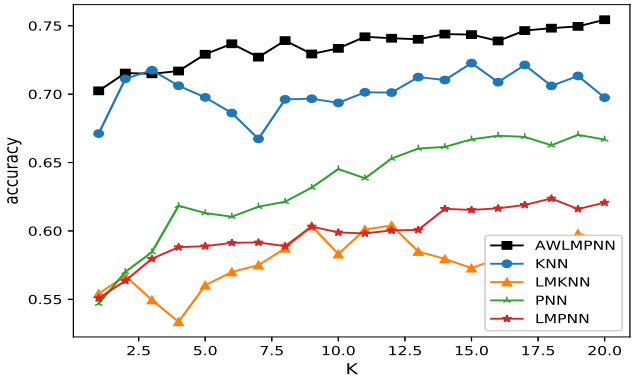
Fig. 2: The classification accuracies of every algorithm via K on all data sets.



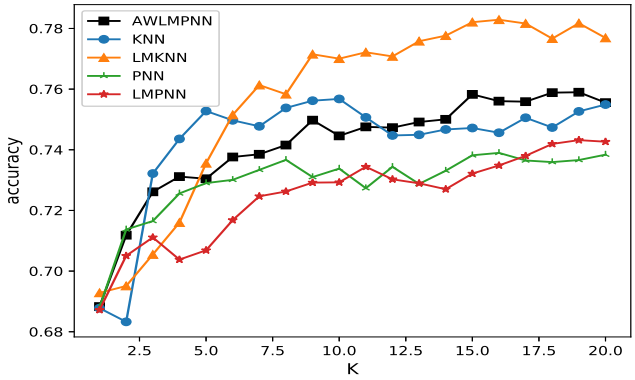
1 Wpbc



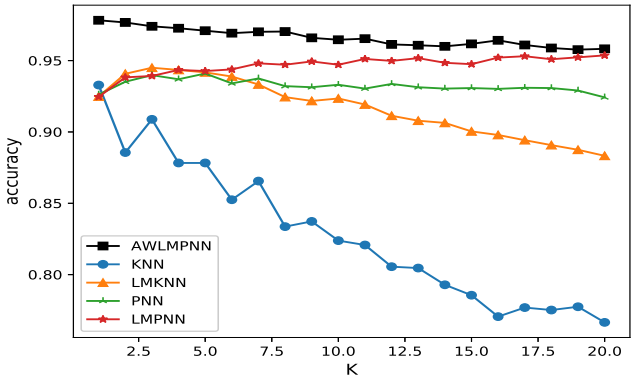
2 Seed



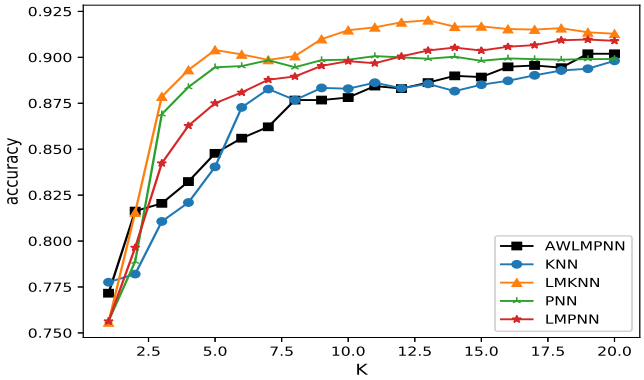
3 Heart



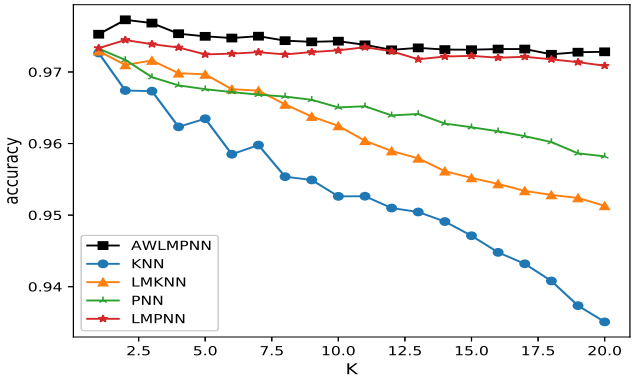
4 Pima



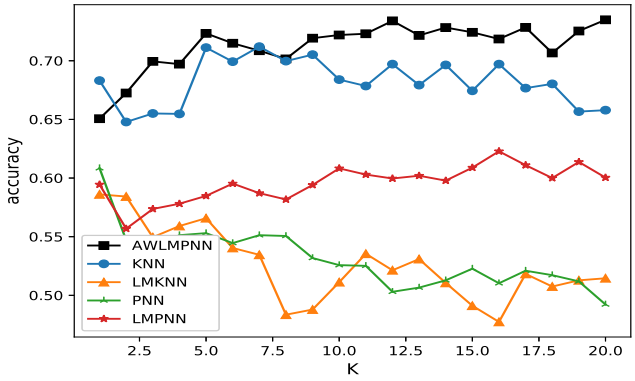
5 Diabetes



6 Balance

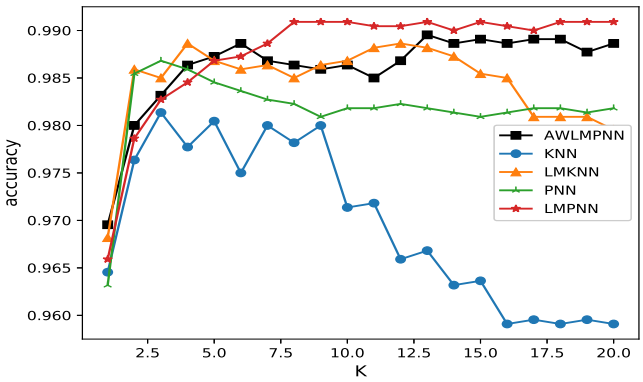


7 Robot

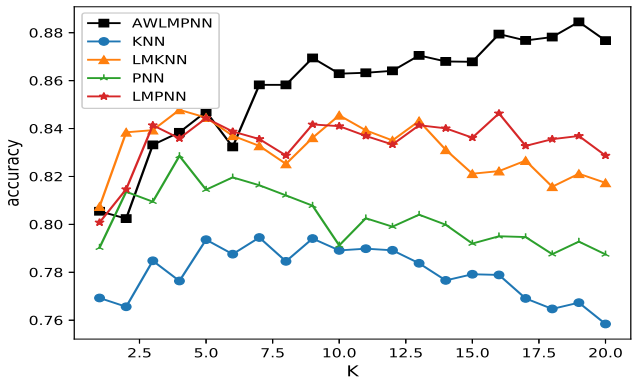


8 Coimbra

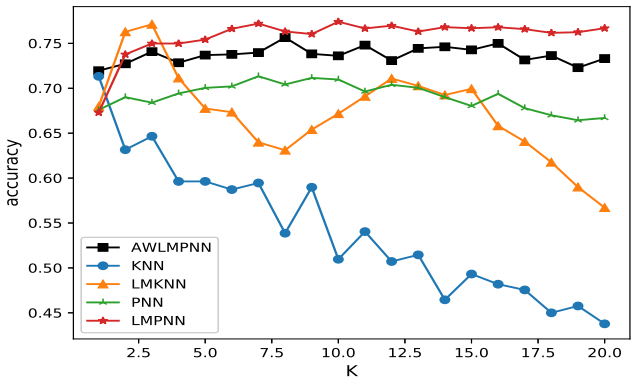
Fig. 3: The classification accuracies of every algorithm via K on all data sets.



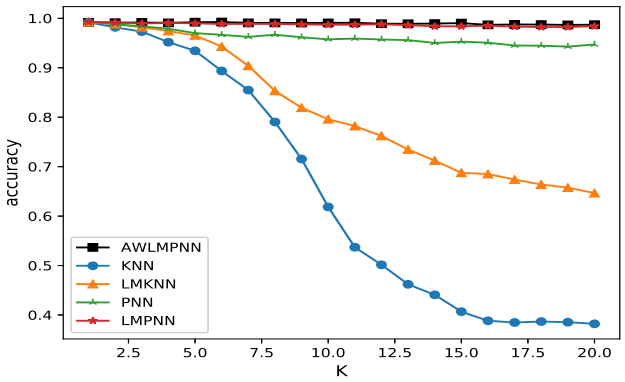
1 Ecoli



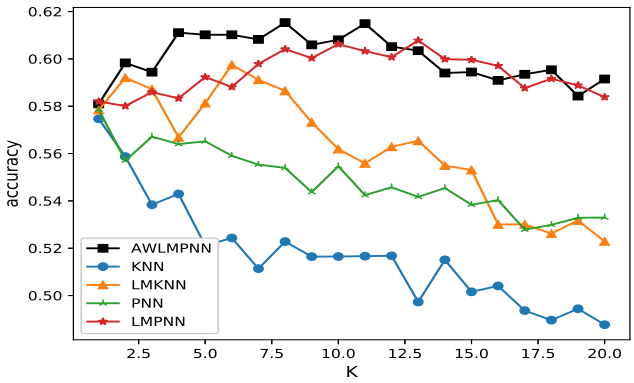
2 Lymphography



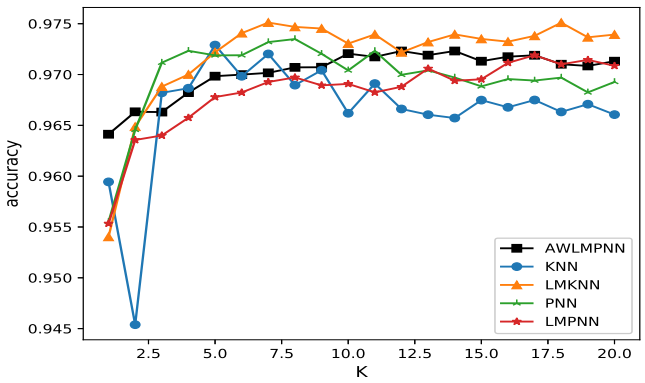
3 Hayesroth



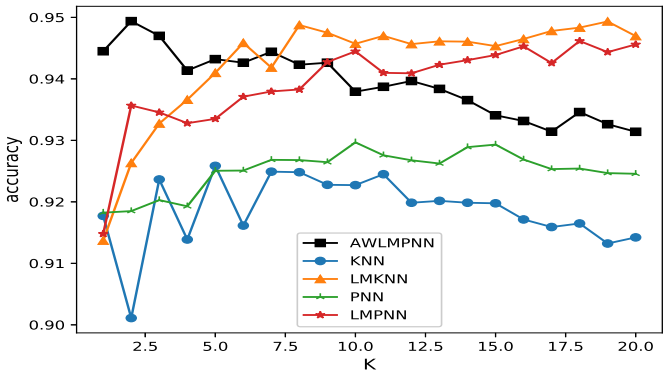
4 Vowel



5 HillValley



6 Breast



7 Housevote

Fig. 4: The classification accuracies of every algorithm via K on all data sets.

TABLE IV: The highest Micro-F1(%) of every algorithm with the relevant stds(standard deviations) and K value in the parentheses.

Data	KNN	LMKNN	PNN	LMPNN	AWLMPNN
Audi	97.33±3.72(1)	97.47±0.95(1)	97.31±1.04(1)	97.64±0.12(13)	**99.84±0.17(2)**
Balance	89.92±3.67(20)	**91.92±3.96(12)**	90.1±3.8(13)	90.9±4.02(20)	90.09±3.44(18)
Bands	71.06±2.97(1)	70.76±3.51(1)	71.98±1.92(2)	71.61±1.04(3)	**77.03±0.53(10)**
Breast	97.41±0.54(5)	**97.5±0.44(8)**	97.25±0.34(4)	97.29±0.37(20)	97.25±0.19(18)
Coimbra	58.89±3.16(1)	59.17±3.72(5)	58.67±2.26(1)	61.46±1.57(17)	**73.07±1.94(16)**
Cryotherapy	91.89±8.39(1)	92.44±9.04(1)	91.89±4.05(1)	93±2.32(2)	**93.44±1.64(18)**
Duser	87.74±1.51(5)	92.61±1.75(9)	90.84±1.49(10)	91.38±1.75(20)	**93.44±1.02(20)**
Dermatology	90.06±4.71(1)	93.25±3.16(4)	91.39±2.3(3)	94.82±1.25(14)	**97.66±0.38(10)**
Diabetes	93.12±4.79(1)	94.65±2(4)	94.13±0.51(3)	95.25±0.65(18)	**97.98±0.67(1)**
Ecoli	98.32±0.83(5)	98.95±0.49(14)	98.59±0.42(4)	**99.09±0.59(12)**	98.95±0.41(16)
Hayesroth	70.97±7.52(1)	76.93±4.78(3)	70.98±1.47(9)	**77.46±2.07(15)**	75.55±1.19(11)
Heart	67.47±2.74(20)	60.39±1.74(12)	66.97±3.48(17)	62.24±1.65(20)	**75.08±1.37(20)**
Hepatitis	64.13±2.89(17)	65±2.25(4)	61.5±1.89(13)	68.63±3.94(19)	**73.13±1.86(14)**
HillValley	57.51±2.32(1)	59.38±2.32(6)	57.99±1.46(1)	60.59±0.74(9)	**61.25±0.96(5)**
Housevote	92.71±0.52(5)	95±0.78(9)	93.06±0.4(14)	94.64±0.8(16)	**95.18±0.56(2)**
Ionosphere	89.63±1.54(2)	90.05±0.94(13)	86.32±0.74(2)	89.52±0.64(20)	**90.86±0.15(16)**
Lymphography	80.15±1.4(11)	84.72±1.13(5)	82.02±0.9(5)	84.26±0.95(7)	**88.14±2.27(17)**
Newthyroid	94.43±2.58(1)	96.14±1.25(3)	94.33±1.49(1)	95.63±0.62(6)	**97.16±0.35(4)**
Parkinsons	84.72±1.46(5)	85.57±2.63(5)	86.81±1.13(7)	86.55±0.81(16)	**95.34±0.29(13)**
Pima	75.52±1.94(9)	**78.29±2.98(17)**	74.09±1.35(15)	74.57±1.58(20)	75.69±1.75(18)
Cleveland	55.8±2.9(20)	55.01±3.75(20)	55.73±3.56(20)	50.67±2.01(17)	**60.94±1.06(20)**
QSAR	82.28±1.29(3)	83.62±1.03(8)	82.62±0.71(11)	83.12±0.86(20)	**86.9±0.94(17)**
Seed	91.14±0.82(9)	91±0.61(2)	90.86±0.55(20)	91.19±0.51(3)	**93.29±0.28(10)**
Robot	97.3±1.02(1)	97.32±0.72(1)	97.28±0.38(1)	97.43±0.08(3)	**97.72±0.13(3)**
Sonar	82.65±5.91(1)	83.17±2.63(4)	83.58±0.44(16)	85.3±0.7(16)	**89.52±0.88(7)**
Tea	60.03±4.98(1)	58.18±4.93(1)	57.54±2.17(1)	58.79±1.98(2)	**63.03±1.3(12)**
Vowel	99.29±24(1)	99.28±12.6(1)	99.15±1.45(1)	**99.3±0.32(1)**	99.28±0.2(6)
Wdbc	93.64±0.62(8)	93.52±0.38(4)	93.41±0.49(13)	93.43±0.53(19)	**97.22±0.41(15)**
Wine	76.57±1.86(1)	76.74±1.44(8)	76.48±2.02(1)	78.35±0.87(17)	**98.65±0.59(15)**
Wpbc	76.16±2.65(20)	72.89±2.33(19)	76.07±3.09(20)	71.07±1.7(20)	**79.16±1.48(19)**
Average	82.27±3.5	83.03±2.67	82.3±1.58	83.17±1.23	**87.06±0.95**

TABLE V: The pairwise contrasts of AWLMPNN with other four algorithms on all data sets employing Wilconxon Signed-Ranks test ('yes' indicate the dramatic difference between two algorithms)

Pairwise comparison	R^+	R^-	Statistics(Z)	P-value	Significant difference
AWLMPNN VS. KNN	456	9	-4.60	4.29×10^{-6}	yes
AWLMPNN VS. LMKNN	427	38	-4.00	6.32×10^{-5}	yes
AWLMPNN VS. PNN	462	3	-4.72	2.35×10^{-6}	yes
AWLMPNN VS. LMPNN	442	23	-4.31	1.64×10^{-5}	yes

$$R^+ = \sum_{\beta_i>0} rank(\beta_i) + \frac{1}{2} \times \sum_{\beta_i=0} rank(\beta_i) \tag{11}$$

$$R^- = \sum_{\beta_i<0} rank(\beta_i) + \frac{1}{2} \times \sum_{\beta_i=0} rank(\beta_i). \tag{12}$$

set $R = min(R^+, R^-)$, if $n > 25$, the statistics is define as

$$Z = \frac{R - \frac{1}{4}n(n+1)}{\sqrt{\frac{1}{24}n(n+1)(2n+1)}}. \tag{13}$$

Z is approximately normal distribution. If $Z < -1.96$, the null hypothesis was rejected at the 5% significance level. Wilconxon Signed-Ranks test is used to compare the proposed AWLMPNN classifier with the other four classifiers in pairs, as shown in table V. It can be seen from table V that R^+ is significantly greater than R^-, and statistical $Z < -1.96$. Therefore, AWLMPNN is significantly outperform than the other four methods. As everyone knows that p-value provides important information for statistical hypothesis testing, which implies that the smaller the p-value value, the more proof to reject the original hypothesis [29]. Obviously, p-value was significantly less than 0.05. In conclusion, the experimental results in table V show that AWLMPNN is superior to KNN, LMKNN, PNN and LMPNN.

TABLE VI: The comparison of multiple classifiers on 30 UCI data sets employing Friedman test.

Methond	KNN	LMKNN	PNN	LMPNN	AWLMPNN
Mean rank	3.9	3.033	3.867	2.8	1.4

Then, Friedman test is used to perform comparative statistical tests on KNN, LMKNN, PNN, LMPNN and AWLMPNN. In each data set, those algorithms are sorted according to its classification performance. After that, the ranking of top algorithm is record as 1, the second is record as 2, and so on. In the case of a draw, an mean ranking is assigned. Let R_i^j be the ranking of the jth of L algorithms on the ith of n data sets, and the mean ranking of jth algorithms are computed as $R_j = \frac{1}{n}\sum_i R_i^j$. Under the null-hypothesis, all the performance of algorithms is similar and so their ranking R^j ought to be equal. The Friedman statistics

$$\chi_F^2 = \frac{12n}{L(L+1)}[\sum_j R_j^2 - \frac{L(L+1)^2}{4}]. \tag{14}$$

is distributed in accordance χ_F^2 with L-1 degrees of freedom

when $n > 10$ and $L \geq 5$.

To further assess the function of the AWLMPNN, we use Friedman test to contrast the classification accuracy of AWLMPNN with the other four algorithms. Their mean ranking of algorithms are display in Table VI. On the basis of Eq. (14), $\chi_F^2 = 49.95$. If the effects of all algorithms are similar, the average ranking is $\overline{R} = 3$. It is obvious from Table VI, the mean ranking R_j of per algorithm is significantly different from $\overline{R}$, and χ_F^2 is greater than $(\chi_F^2)_{0.05} = 9.49$. Hence, there exist clear distinguish among these five algorithms.

To illustrate the remarkable of Friedman test, We used Holm post hoc test to statistically contrast AWLMPNN with other four algorithms. The pairwise contrast statistics of AWLMPNN with other algorithms is calculate as

$$Z = (R_i - R_0)/SE \tag{15}$$

where R_0 and R_i are the mean ranking of AWLMPNN and the ith of the other four algorithms in Table VI, respectively. SE is obtained from formula $\sqrt{\frac{L(L+1}{6n}} = \sqrt{\frac{6\times(6+1)}{6\times30}} = 0.408$. Z value is applied to obtain the relevant probability, and then contrasted with appropriate $\alpha = 0.05$. Table VII displayed the sorted statistics and corresponding p-value. As revealed in Table VII, because the relevant p-value are much less than the adjusted $\alpha's$. AS a result, four original assumptions are rejected, which means that the AWLMPNN is remarkably better than other existing algorithms.

TABLE VII: The $p-$value in the Friedman test employing the Holm post hoc test.

i	Method	$Z = (R_i - R_0)/SE$	P-value	$\alpha/(k-i)$
1	KNN	6.127	9×10^{-10}	0.0125
2	PNN	6.047	1.5×10^{-9}	0.0167
3	LMKNN	4	6.3×10^{-5}	0.025
4	LMPNN	3.341	6×10^{-4}	0.05

Finally, according to the results in table II, T-test was used to compare AWLMPNN with KNN, LMKNN, PNN and LMPNN. The results are shown in table VIII. Obviously, the p-value is less than 0.01. Therefore, the experimental results in table VIII show that the proposed AWLMPNN is significantly different from KNN, LMKNN, PNN and LMPNN.

TABLE VIII: The pairwise contrasts of AWLMPNN with other four algorithms on all data sets employing T-test ('yes' indicate the dramatic difference between two algorithms)

Pairwise comparison	P-value	Significant difference
KNN VS. AWLMPNN	3.24×10^{-5}	yes
LMKNN VS. AWLMPNN	3.41×10^{-4}	yes
PNN VS. AWLMPNN	3.38×10^{-6}	yes
LMPNN VS. AWLMPNN	9.97×10^{-5}	yes

In a word, the proposed AWLMPNN classification performance is widely verified in real data sets. The results can be summarized as follows:

- The effectiveness and robustness of AWLMPNN have been verified on a large number of real data sets, and it is a promising pattern recognition classifier.
- In the case of high dimensional attributes with small sample size, AWLMPNN has better classification performance than other algorithms.
- The proposed AWLMPNN can overcome the sensitivity of neighborhood size K and has good classification performance.
- Compared with Euclidean distance, the attribute weighting distance used in AWLMPNN can obtain more suitable nearest neighbors and make favorable classification decisions.

V. Conclusion

In this paper, we introduce a new KNN-based rule AWLMPNN. The main purpose of AWLMPNN is to overcome the influence of non information features and further improve the classification performance. In AWLMPNN, an attribute weighting distance is designed, which fully considers the classification contribution of different attributes, so as to find a more suitable nearest neighbor for classification. To highlight the classification performance of this method, we conducted comprehensive experiments on 30 real UCI data sets, and compared with LMKNN, PNN, LMPNN and KNN classifiers. Experimental results show that the proposed AWLMPNN is effective and robust, and satisfactory classification results are obtained. In the future work, we plan to adaptively select the value of neighborhood size K in KNN-based classification, and design an adaptive weighting method for K nearest neighbors or local mean vectors to further improve the classification performance.

References

[1] T. M. Cover, P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, 13 (1967): 21-27.

[2] X. Wu, V. Kumar, J. R. Quinlan, et al, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, 14.1(2008):1-37.

[3] J. Gou, L. Du, Y. Zhang, et al, "A New Distance-weighted k-nearest Neighbor Classifier," *Journal of Information & Computational Science*, 9.6(2012).

[4] S. Okamoto, K. Satoh, "An Average-Case Analysis of k-Nearest Neighbor Classifier," *Case-Based Reasoning Research and Development, First International Conference, ICCBR-95, Sesimbra, Portugal*, October 23-26(1995).

[5] Y. Li, Y. Yang, J. Che, L. Zhang, "Predicting the number of nearest neighbor for kNN classifier," *IAENG Int J Comput Sci*, 46.4 (2019): 662-669.

[6] J. Wang, P. Neskovic, L. N. Cooper, "Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence," *Pattern Recognition*, 39.3(2006):417-423.

[7] F. Bulut, M. F. Amasyali, "Locally adaptive k parameter selection for nearest neighbor classifier: one nearest cluster," *Pattern Analysis and Applications*, (2015):1-11.

[8] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Trans Systems Man & Cybernetics*, 6.4(1976):325-327.

[9] J. Gou, T. Xiong, Y. Kuang, "A Novel Weighted Voting for K-Nearest Neighbor Rule," *Journal of Computers*, 6.5(2011):833-840.

[10] M. Bicego, M. Loog, "Weighted K-Nearest Neighbor revisited." *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, (2016).

[11] P. Li, J. Gou, H. Yang, "The distance-weighted K-nearest centroid neighbor classification," *Journal of Information Hiding and Multimedia Signal Processing*, 8.3(2017):611-622.

[12] C. Domeniconi, J. Peng, D. Gunopulos, "Locally adaptive metric nearest-neighbor classification," *Pattern Analysis & Machine Intelligence IEEE Transactions on*, 24.9(2002):1281-1285.

[13] J. Wang, P. Neskovic, L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognition Letters*, 28. 2(2007):207-213.

[14] K. Q. Weinberger, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Jmlr*, 10(2009).

[15] Y. Xu, Q. Zhu, Z. Fan, et al, "Coarse to fine K nearest neighbor classifier," *Pattern Recognition Letters*, 34.9(2013):980-986.

[16] W. Li, Q. Du, F. Zhang, et al, "Collaborative-Representation-Based Nearest Neighbor Classifier for Hyperspectral Imagery," *IEEE Geoscience and Remote Sensing Letters*, 12.2(2014):389-393.

[17] J. Gou, H. Ma, W. Ou, et al, "A generalized mean distance-based k-nearest neighbor classifier," *Expert Systems with Applications*, 115(2018).
[18] L. Chen, G. Guo, "Nearest neighbor classification of categorical data by attributes weighting," *Expert Systems with Applications*, 42.6(2015):3142-3149.
[19] X. Cui, D. Xu, H. Xian, et al, "A Novel TOPSIS-MABAC Method for Multi-attribute Decision Making with Interval Neutrosophic Set." *IAENG International Journal of Applied Mathematics*, 50.2(2020):273-741.
[20] J. Lopez, S. Maldonado, "Redefining nearest neighbor classification in high-dimensional settings," *Pattern Recognition Letters*, 110.jul.15(2018):36-43.
[21] A. Gul, A. Perperoglou, Z. Khan, et al, "Ensemble of a subset of kNN classifiers," *Advances in Data Analysis & Classification*, 12.4(2016):1-14.
[22] Y. Mitani, Y. Hamamoto, "A local mean-based nonparametric classifier," *Pattern Recognition Letters*, 27.10(2006):1151-1159.
[23] Y. Zeng, Y. Yang, L. Zhao, "Pseudo nearest neighbor rule for pattern classification," *Expert Systems with Applications*, 36.2p2(2008):3587-3595.
[24] J. Gou, Y. Zhan, Y. Rao, et al, "Improved pseudo nearest neighbor classification," *Knowledge-Based Systems*, 70(2014):361-375.
[25] B. Li, Y. W. Chen, Y. Q. Chen, "The Nearest Neighbor Algorithm of Local Probability Centers," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38.1(2008):141-154.
[26] A. A. R. Asif, M. M. Nishat, F. Faisal, et al, "Performance Evaluation and Comparative Analysis of Different Machine Learning Algorithms in Predicting Cardiovascular Disease," *Engineering Letters*, 29.2(2021):731-741.
[27] K. Bache, M. Lichman, UCI Machine Learning Repository <http://archive.ics.uci.edu/ml> University of California, School of Information and Computer Science (2013), Irvine, CA.
[28] J. Derrac, S. Garcła, D. Molina, et al, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm & Evolutionary Computation* , 1. 1(2011):3-18.
[29] I. Salvador Garc, F. Herrera, "An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons," *Journal of Machine Learning Research*, 9.12(2008):2677-2694.
[30] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, 1.6(1945):80-83.
[31] J. Demiar, D. Schuurmans, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, 7.1(2006):1-30.