IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Medical Health Big Data Classification Based on KNN Classification Algorithm

Wenchao Xing [1], Yilin Bei[2]*

1.School of Primary education；JiNing University; Qufu Shandong; 273100; China
2.School of Information Science and Technology; TaiShan University; Taian Shandong;271000; China

Corresponding author: Yilin Bei (e-mail: beiyilinok@163.com).

ABSTRACT The rapid development of information technology has led to the development of medical informatization in the direction of intelligence. Medical health big data provides a basic data resource guarantee for medical service intelligence and smart healthcare. The classification of medical health big data is of great significance for the intelligentization of medical information. Due to the simplicity of KNN (K-Nearest Neighbor) classification algorithm, it has been widely used in many fields. However, when the sample size is large and the feature attributes are large, the efficiency of the KNN algorithm classification will be greatly reduced. This paper proposes an improved KNN algorithm and compares it with the traditional KNN algorithm. The classification is performed in the query instance neighborhood of the conventional KNN classifier, and weights are assigned to each class. The algorithm considers the class distribution around the query instance to ensure that the assigned weight does not adversely affect the outliers. Aiming at the shortcomings of traditional KNN algorithm in processing large data sets, this paper proposes an improved KNN algorithm based on cluster denoising and density cropping. The algorithm performs denoising processing by clustering, and improves the classification efficiency of KNN algorithm by speeding up the search speed of K-nearest neighbors, while maintaining the classification accuracy of KNN algorithm. The experimental results show that the proposed algorithm can effectively improve the classification efficiency of KNN algorithm in processing large data sets, and maintain the classification accuracy of KNN algorithm well, and has good classification performance.

INDEX TERMS improved KNN classifier; weighted KNN algorithm; cluster denoising; density cropping.

I. INTRODUCTION

The development of information technology has made digital medical technology more mature, medical data is growing at an unprecedented rate, and biomedical research has developed into a typical data-intensive science, forming a data explosion phenomenon called "big data." In the era of big data, data has become a new strategic resource, an important factor driving innovation, and is changing the way of biomedical research and the way of life and thinking of human beings. Through the integration analysis and application requirements description of big data in the medical service field, relevant departments of the medical industry can be guided to strengthen the collection and management of medical health big data, and lay a data foundation for later development and application [1-3]. At the same time, it provides a good theoretical and technical basis for the development and utilization of medical health big data. The key technologies and data models of medical health big data provided by the project research results can enrich the theoretical system and application system in the field of medical health big data research [4,5].

Many developed countries have successively released a series of big data technology plans, vigorously promoted big data research and application, and have regarded health care and health big data as a key component of national public utilities [6,7]. The use of health data has become a new indicator of the country's economic development. The University of Oxford has established a Health Information and Discovery Center. The medical and health research center has integrated big data technology and has two research institutes, the Big Data Research Institute and the Target Discovery Research Institute [8]. Classification techniques are based on inductive learning principles that analyze and find patterns from databases [9]. If the nature of the environment is dynamic, then the model must be adaptive. It should be able to learn and map effectively. Related scholars have proposed a regenerative space framework for information theory learning [10-12]. The framework uses a symmetric non-negative definite kernel function, the potential for cross information. Although this framework gives better results than the previous RKHS framework, there is still the problem of choosing the appropriate kernel

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

function for a particular domain [13]. Relevant scholars have combined fuzzy and rough set theory to obtain the smallest subset of eigenvalues in noise and real data [14-16]. It provides greater flexibility when dealing with real data and noise data. Other feature ranking metrics are also used to evaluate new coarse metrics [17-18]. This measure gives reliable results comparable to others. Fuzzy and ant colony optimization are two new methods to introduce an enhanced feature selection framework [19-21]. The related scholars brought the maximum entropy model into the research of text automatic classification system for the first time [22,23]. Through the simulation experiment, the different entropy model based classifiers were analyzed in different text feature generation methods. The researchers proposed a new text feature selection algorithm based on Gini index [24]. The feature selection algorithm uses the Gini index principle to study text feature selection, and constructs a feature selection evaluation function suitable for text classification feature selection based on Gini index. By eliminating the noise data of training samples and the training samples with high similarity, the training samples are reduced to improve the classification efficiency [25,26]. However, such algorithms are susceptible to improper threshold setting and uneven distribution of training sample categories, resulting in poor classification results [27]. By introducing the scaled convex hull method, the maximum boundary value algorithm for cost-sensitive learning can be solved and applied to the problem of unbalanced data. The error rate of a few class classifications is reduced by changing the scale factor of the scaled convex hull method. In applying the cost-sensitive learning decision tree to medical diagnosis, new test strategies can be used to reduce misclassified samples. However, the biggest disadvantage of cost-sensitive learning is that it is difficult to determine the value of the generation. If the value of the generation is not well determined, it will not help the improvement of the algorithm.

With the gradual growth of medical and health care data, the traditional medical health big data classification methods have problems such as large sample size and slow processing. In order to better classify the unbalanced data set, the K nearest neighbor algorithm is modified. A theoretical model of the KNN (K-Nearest Neighbor) algorithm is established. This model does not need to consider the data around a certain data, but assigns the weight of each category according to the location. By considering the distribution of the classes, the accuracy of the weights for the special values is guaranteed. In this way, the traditional KNN classifier has been improved. In addition, considering the time lag of the traditional KNN algorithm in the face of large data sets, this paper proposes to improve the KNN algorithm by cluster denoising and density clipping. The results show that the classification speed is improved and the classification accuracy is guaranteed.

Specifically, the technical contributions of this paper can be summarized as follows:

First: An experimental study was conducted on the class-based weighting factors of the KNN classifier. A class-based weighted K nearest neighbor algorithm is proposed and the attributes of the associated weighting factors are viewed. We evaluate methods for various real-world data sets and compare their performance to the most advanced classifiers.

Second: An improved KNN algorithm based on DBSCAN cluster denoising and density cropping is proposed. The clustering method is used to study the training set samples, and the density of each cluster is used to speed up the search speed of K-nearest neighbors. The classification efficiency of KNN algorithm in processing large data sets is improved, and the classification accuracy of KNN algorithm is maintained.

The rest of this paper is organized as follows. Section 2 discusses related theories and methods, Section 3 analyzes the class-based weighted K nearest neighbor algorithm, and Section 4 studies the improved KNN algorithm based on the clustering algorithm. Section 5 summarizes the full text.

II. RELATED THEORIES AND METHODS

A. Medical health big data technology classification

Health care big data, like most industry big data, has a similar process. Big data technology is generally divided into data acquisition, data storage, data analysis and data display. The medical health big data life cycle model is shown in Figure 1.

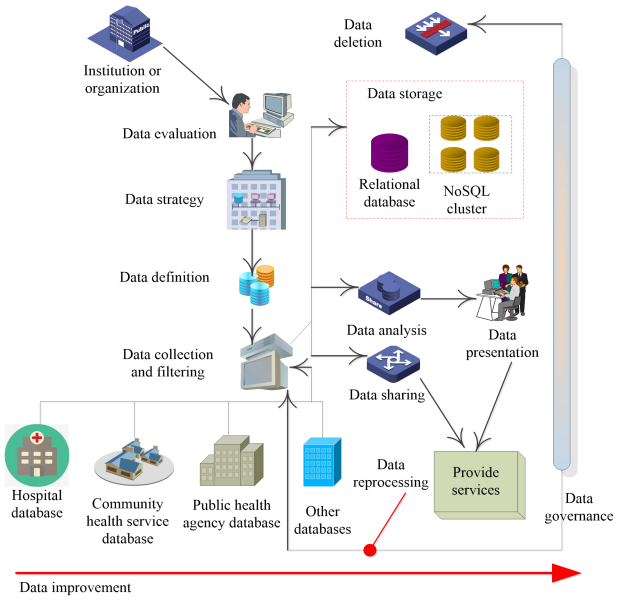


FIGURE 1.**Medical health big data life cycle model**

For small projects, the source of the data is single, the amount of data is small, and the traditional relational database technology or data warehouse can fully meet the corresponding needs. Because of the huge amount of data in big data, it has high requirements for storage, analysis, and display. With distributed storage computing, data security and timeliness should be more focused. The following are big data relevant technologies. Several main processes are described:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

1) Big data acquisition technology

There are a large number of unstructured, semi-structured and structured data in medical health big data, and big data acquisition technology is to obtain this part of data through various means. It is the starting point of big data mining. The distributed system quickly and reliably captures data, realizes data parsing, conversion and loading, and provides a secure and consistent big data collection technology.

2) Big data storage and management technology

This part is to solve the data storage and management, and to provide a big data storage technology, such as HDFS, which is safe, efficient, low cost and highly fault tolerant. Moreover, due to the unstructured data, it is generally necessary to have big data management and storage capabilities for unstructured data.

3) Big data application technology

The above three processes are generally only related to the data mining personnel, and do not deal with the user, but the data mining results presented to the user is the final stage of the data mining process, that is, various big data visualization technologies. This part of the application technology is versatile and there is no uniform standard.

B. Regression algorithm and KNN classification algorithm

1) Statistical methods

The statistical method follows the "curve fitting" method and requires the form of a curve to be predetermined. They try to model the relationship between dependencies and independent variables as closure functions. It is the responsibility of the user to make an informed guess about this feature. This is done by studying the application domain and may involve trial and error. Once the functional form is modified, its parameters (or coefficients) are estimated to be "best fit" to the available data. Typically, this function also has an error term for compensating for unexplained changes in the dependent variables. This requires that the regression problem for each particular application area can be best studied and solved in this area. Changes in linear regression include pace regression, minimum median regression, and so on.

2) Decision tree stepwise regression

The basic idea of the stepwise regression algorithm is to introduce the independent variables one by one, and the independent variables introduced each time have the most significant influence on the dependent variable Y. Each time a new independent variable is introduced, the old independent variables previously introduced into the regression equation are tested one by one, and the insignificant independent variables in the current equation are removed from the independent variables with the least influence on the dependent variable Y, and are eliminated one by one, until you can no longer introduce new arguments. Finally, the independent variables retained in the regression equation have a significant influence on the dependent variable Y, and the independent variables in the regression equation are not significant to Y. Such a regression equation is called the optimal regression equation.

With m independent variables and n observations, a linear regression model of m is obtained:

$$y_i = \beta_0 + \beta_1 x_{ij} + \beta_j x_{ij} + \beta_m x_{im} + e_i \quad (1)$$

Where i = 1, 2, ..., n represents the i-th observation, j = 1, 2,..., m represents the j-th argument, β_j is the partial regression coefficient, and e_i is the random error.

3) KNN classification algorithm

The KNN algorithm is a generalization algorithm for nearest neighbor rules. Its inductive offset is the class label of the k-sample with the class label to be tested most similar to the nearest one. Compared with the nearest neighbor, it differs in that it expands the nearest neighbor to k in the decision-making phase. This extension allows the KNN algorithm to obtain and utilize more information. It omits the process of learning processing relative to other classification algorithms with distinct training phases. The KNN classification diagram is shown in Figure 2.

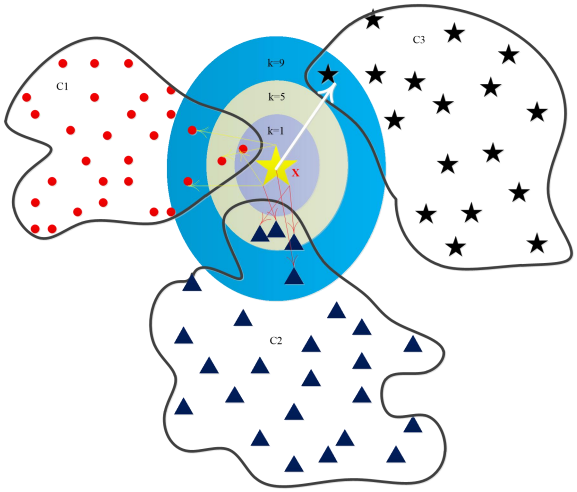


FIGURE 2.**Schematic diagram of KNN classification**

The nearest neighbor rule is one of the oldest methods of class reasoning. Its decision-making idea is very simple, that is, the sample to be tested is the same as the sample category closest to it. If the training set and the distance metric are kept unchanged, the decision result of the nearest neighbor rule has been uniquely determined for any instance to be tested. For all sample instances in set E, if y is the nearest neighbor instance of x, then the category of y is the result of the decision, which is the nearest neighbor rule. Let X be an unknown category sample, the specific decision process is:

$$g_j(X) = \min g_i(\mathrm{X}) \quad i = 1,2,...,C \quad (2)$$

Then the decision result is $X \in W_i$.

Here, the nearest neighbor rule is introduced from the following two aspects: one is convergence, and the other is generalization error.

For the same point to be tested x, the nearest neighbor x' obtained by using two training sets containing different samples is different. Since the classification result depends

on the category label of the nearest neighbor, $P(e|x,x')$ is thus obtained. It is the conditional error rate and depends on both x and x'. Here, the average of x' can be obtained:

$$P(e|x) = \int P(e|x,x')p(x'|x)dx' \quad (3)$$

Among them, $p(x'|x)$ is a conditional probability density function. Assuming that $p(\cdot)$ is a continuous non-zero function, the probability that any point falls on the x-centered hypersphere S is:

$$P_s = \int p(x')dx' \quad (4)$$

The probability that all n samples fall outside the hypersphere is $(1-P_s)^*$. If $n \to \infty$, the probability tends to zero. It can be concluded that if the nearest neighbor x' converges to the point x to be measured according to the probability, then $P(e|x)$ infinitely approaches the Dirac function. Similarly, if the KNN decision rule is followed, then k neighbors converge to the point x to be measured.

The error rate of the nearest neighbor can be understood as the probability that the point to be measured x is different from the category c of the nearest neighbor point x', and the error rate is:

$$P(error) = 1 - \sum_{c \in Y} P(c|x)P(c|x') \quad (5)$$

Here, the assumption is made that each sample is independently and equally distributed. A sample x can always be found within the d distance range around x, so that the Bayesian classifier is:

$$c^* = \arg\max_{c \in Y} P(c|x) \quad (6)$$

At this time there are:

$$P(error) \le 1 - \Sigma_{c \in Y} P^2(c^*|x) \quad (7)$$

It is concluded that the nearest neighbor rule is not only simple in construction, but also the generalized error rate is not more than twice the Bayesian error rate.

III. CLASS-BASED WEIGHTED K NEAREST NEIGHBOR ALGORITHM

A. Research on weighting factors

1) Design of class-based weighting factors for KNN classifiers

The simple basic design of the KNN classifier based on the weighting factors of the class can be expressed as follows:

$$W[c] = \frac{1}{frequency[c]} \quad (8)$$

$W[c]$ represents the weighting factor of class C. In addition, $frequency[c]$ represents the frequency of occurrence of class C in the entire data. If the data is perfectly balanced, this means that the $frequency[c]$ values of all classes are roughly equal, so the modified KNN algorithm reduces the balanced data to the conventional KNN algorithm. For a given query instance x_t, the modified KNN rule can be formally expressed as follows:

$$y_t = \arg\max \Sigma W[c] * E(y_i, c) \quad (9)$$

This weighting factor has major drawbacks and does not perform the same on most data sets. It can be seen from the experimental research in this paper. This design only takes into account the overall imbalance between the data, regardless of the distribution of local data. When data points are clustered, each cluster has a main class and the class distribution is unbalanced. In this case, the algorithm tends to be less than other categories. This situation is shown in Figure 3.

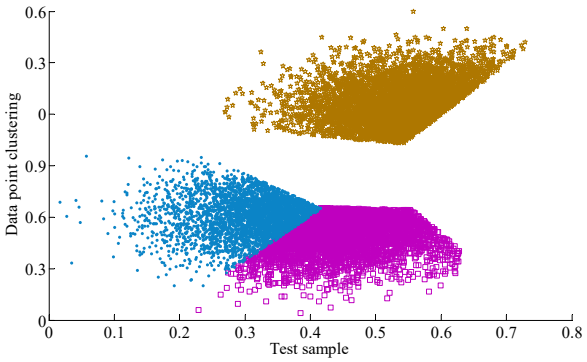


FIGURE 3. **Sample test results**

2) Improved design of KNN classifier

Since the earlier proposed design did not perform well on most data sets and had major drawbacks, corresponding adjustments were needed. In the new proposed design, this paper introduces a coefficient based on the unbalanced nature of the data set. Therefore, a modified KNN rule for a given query instance x_t can be formally expressed as:

$$y_t = \arg\max \Sigma W[c] * (1 + E(y_t, c)) \quad (10)$$

Where W[c] represents the weighting factor of class c and can be defined as:

$$W[c] = \frac{1}{1 + \alpha * (\frac{frequency[c]}{\Sigma_{t=1}^{m} frequency[c_j]})} \quad (11)$$

Where α is an input parameter that can be used as input from the user or learned from the data. $\frac{frequency[c]}{\Sigma_{t=1}^{m} frequency[c_j]}$ is the ratio of the occurrence rate of class c in the entire data to the total number of instances in the entire data.

For perfectly balanced data, the value of a should be zero, which will cause all W[c] values to be equal to 1, so the modified KNN algorithm on the balanced data is reduced to the existing KNN algorithm. The performance of the improved KNN classifier is very sensitive to the value of a. The alpha factor attempts to take into account the nature of the data distribution, which was ignored in earlier designs. However, the current design also fails to capture the local class distribution around the query instance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

The weighting factor is between 1 and $\frac{frequency[c]}{1+\alpha * \sum_{i=1}^{m} frequency[c_j]}$. The results of testing the text classification quality are shown in Figure 4. The test uses medical health data of different lengths and categories.

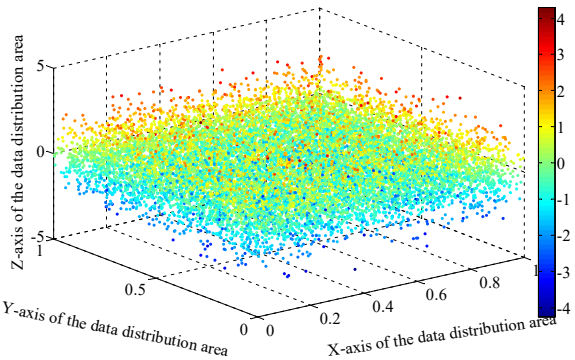


FIGURE 4.**Sample detection three-dimensional distribution**

3) Consider only the design of the area near the query instance

Only the area near the query instance is considered here, instead of considering the nature of the entire data. More clearly, if k is the number of neighbors used by the existing KNN algorithm to determine the query instance class, then in this design, this paper considers the class distribution in the k+d nearest neighbors of the query instance. In previous designs, too many remote instances also had an impact when categorizing query instances, but in this one, we try to limit the region so that instances within that region only affect the decision. Therefore, for a given query instance x_t, the modified KNN algorithm can be formally expressed as follows:

$$y_t = \underset{c\in\{c_1,c_2,\ldots,c_m\}x_t\in N(x_t,k)}{\arg\max} \Sigma W(c,x_t) * E(y_t,c) \tag{12}$$

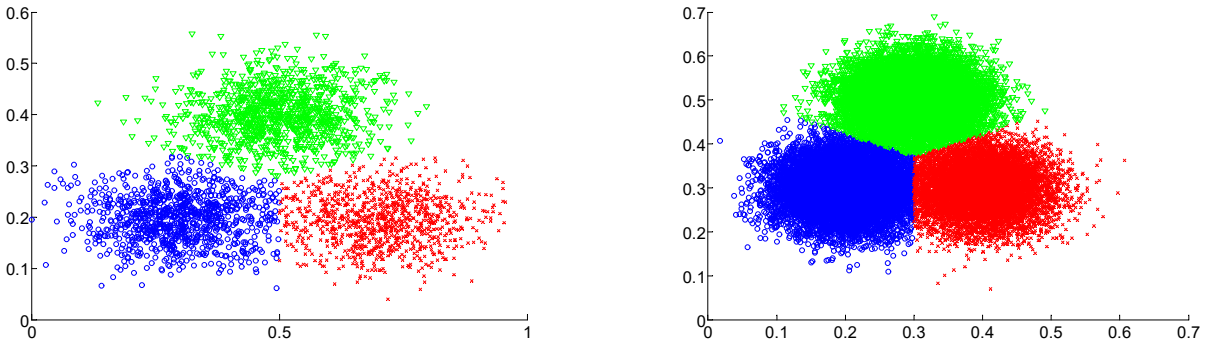
Where $W(c,x_t)$ represents the weighting factor of class c and instance x_t, which can be defined as:

$$W(c,x_t) = \frac{1}{\sum\limits_{x_t\in N(x_t,k+d)} E(y_t,c)} \tag{13}$$

Where d is the input parameter, which can be used as input from the user or learned from the data. $\sum\limits_{x_t\in N(x_t,k+d)} E(y_t,c)$ is the number of instances of class c that exist in the nearest neighbor of x. The algorithm will more closely monitor the nature of the data around the query instance. The results show that the algorithm is successfully implemented in the framework of the classification quality with the increase of the number of files. As we predicted, other tests and analyses showed better results if files belonging to the same category were processed.

4) Simulation experiment

Figure 5 shows the classification results of the KNN algorithm before and after the improvement.



(a) Classification results before improvement **(b) Improved classification results**

FIGURE 5.**KNN algorithm classification results**

Comparing Figure 5(a) with Figure 5(b), we can see that the improvement has a greater impact on the accuracy of the classification. The classification results of the KNN algorithm before and after the improvement are significantly better than those before the improvement.

B. Improved KNN based on feature weight correction

All the features of the feature vector are sorted according to the sensitivity from high to low, and then the features of the partial sorting are deleted, and then the remaining features are trained by the neural network, and the obtained error continues to be deleted within the allowable range. Such remaining features are features that are important to the classification.

In order to reduce the computational complexity of the neural network in the feature deletion work, this section adopts a new neural network feature selection method, which is referred to as the "two-point selection method" in this paper. First, the features are sorted according to the sensitivity, and then a feature R is found by the dichotomy. With the feature R as the boundary, all the features that are listed are deleted. This method greatly reduces the amount of computation.

Since the nearest neighbor classifier assumes that the conditional probability of the local class is constant, this assumption is invalid in the high dimensional feature space. Therefore, the K nearest neighbor classifier is used in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

high dimensional feature space, and the correction of the feature weight will cause serious deviation.

In order to reduce the time of finding the initial k_0 nearest neighbors in the classification, the SS-Tree method is used to divide the training sample space into several small regions according to the similarity between samples. Then, according to the distance of the sample x to be classified from the center point of each region, several regions are first found, and k_0 nearest neighbor samples are found in these regions as approximate k_0 nearest neighbors of x.

In the SS-Tree section of this section, each node represents an area, including the center point, the radius, the sign of reinsertion, the nature of the child (node or sample), the total number of children, and an array of pointers to the child. The sample data is incremented by a flag that is reinserted.

In the process of establishing SS-Tree, all training samples are first chained into a sample queue to be inserted, and an empty tree is initialized. Then the samples are inserted into the SS-Tree one by one. The most important thing in this tree building process is the algorithm of selecting the insertion node of the sample, and the node splitting algorithm to keep the number of children in each node not exceeding B.

In this section, the criterion for selecting the insertion node is the node whose center point is closest to the sample to be inserted. The algorithm select_node of the node selection process in the SS-Tree construction process is:

Algorithm select-node:
(1) t ← the root node of SS-Tree;
x ← to be inserted into the sample;
(2) if (t children are not samples)
Do
{
calculate the distance between the center point of each child of x and t;
t←distance x the nearest child;
}
while(t child is a sample)

After confirming that node x should be inserted in SS-Tree, it inserts x as a child of t. Among them, the insert node algorithm insert is:

Algorithm insert:
If(t is empty) // empty tree
{
establish a root node;
x insert the root node;
}
Else if(t number of children <B)
{
Insert x into t;
Update the individual attribute values of t;
}
Else if (t children have been reinserted)
{
insert x into t ;
Split(t);// split node
}
Else
{
Insert all samples of the t-node into the sample queue to be inserted and mark the re-insertion flags of these samples;
Delete(t);
}

In order to keep the number of children at each node not exceeding B, when the number of children at the node exceeds B, the node is split, and the splitting algorithm is critical to the performance of SS-Tree. In the SS-Tree in this section, the node splitting is performed with the principle of minimizing the area sum of the two regions after splitting. The algorithm for splitting the node t is:

Algorithm split (t):
Every child for t y
{
For i = b to (B+1-b)
{
The distance i is a group of i children, and the remaining nodes are a group;
Calculate the sum of the areas of the two groups s;
}//for select the record with the smallest s value and its s;
}//for selects the smallest group of values s, the group consists of two subsets group1 and group2, according to which the node t is divided into two nodes tg1 and tg2, and the respective attribute values of the two nodes are calculated;
If (t is the root node)
{
Clear the child records of the root node;
Insert children with nodes tg1 and tg2 as root nodes;
Recalculate each attribute value of the root node;
}
Else
{
Remove the record of t in the original t parent node;
Insert the split two nodes tg1 and tg2;
If (t parent node ≥ B)
Split (t parent node);
Else
Update each attribute value of the t parent node;
}

During the establishment of the SS-Tree, when a node is reinserted, the node is deleted first, and the number of children of the parent node of the node may be less than b. In order to keep the number of children at each node not less than b, the node consolidation work is carried out, with the minimum area increase after the merger as the criterion.

After the SS-Tree is established, the area division of the node for the sample is the division of the training sample set that is sought, and the number of samples in each area is between [b, B]. In order to record these area divisions, this area is divided into a table. In order to reduce the time to find the sample boot disk later, the training samples are rearranged in the order shown in this table and then stored.

C. Experimental research

To adjust the existing KNN algorithm, a weighting factor is introduced for each class. For a given query instance x_t, the algorithm can be formally expressed as follows:

$$y_t = \arg\max \Sigma W[c, x_t] * E(y_1, c) \tag{14}$$

Where $W[c, x_t]$ represents the weighting factor for class c, and the query instance x_t is classified.

In order to design weights, query dependencies and query independent weighting factors are considered. If the weighting factors studied in this paper have invariant values for each category of dataset, ie they are not dependent on the query instance and are beneficial to a few classes, then the algorithm under study will exist in the correct cluster, but there is no correct classification. The weight factor value $W[c, x_t]$ of the study can be expressed as:

$$W(c, x_t) = \frac{\alpha(c, x_t)}{1 + \alpha(c, x_t)} \tag{15}$$

Among them,

$$\alpha(c, x_t) = \Sigma \frac{m * getcoef(x_t)}{m} \tag{16}$$

Due to the leading role of most classes, the use of overall accuracy is not an appropriate measure of the imbalanced data set, so this paper uses F-Score as an indicator. F-Score considers the accuracy and recall of the test to calculate the score. Table 1 compares the results of the modified algorithm with the prior art algorithm. The numbers in parentheses indicate the level of the corresponding algorithm. It can be seen that the research method produces consistent and accurate classifiers in most data sets and is superior to other algorithms. Furthermore, the algorithm proposed in this paper is always superior to the conventional KNN on all data sets, which confirms that the modified KNN algorithm is classified according to the nature of the data.

TABLE 1
Experimental results of several data sets

Dataset	Naive	KENN	KNN	Method of this paper
1	0	0	0	0
2	0.49	0.48	0.46	0.56
3	0.68	0.65	0.61	0.72
4	0.72	0.68	0.66	0.73
5	0.77	0.73	0.67	0.76
6	0.79	0.77	0.71	0.82
7	0.81	0.79	0.75	0.83
8	0.82	0.81	0.78	0.84
9	0.83	0.81	0.78	0.85
10	0.83	0.81	0.78	0.85

Figure 6 compares the performance of the studied algorithm with the KNN algorithm in terms of overall accuracy and accuracy. A small number of data is classified according to different k values of the data set. As can be seen from the figure, the algorithm-based classifier is more sensitive to classifying a few data classes and is still highly accurate. The classifiers learned from the research method are more accurate for larger k values.

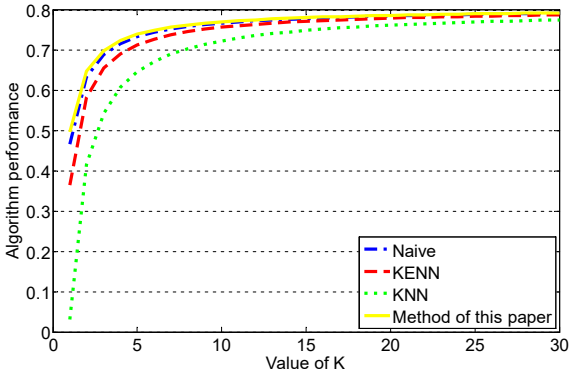


FIGURE 6. **Performance comparison between the algorithm studied and the KNN algorithm**

IV. IMPROVED KNN ALGORITHM BASED ON CLUSTERING ALGORITHM

A. KNN algorithm similarity calculation method

The principle of the KNN classification algorithm is to calculate the similarity of the test text to each training text, and then sort according to the similarity size, select the top K training texts with the highest similarity to the test text, and divide the test text into the K texts. Common similarity calculation methods are generally Euclidean distance and cosine of the angle.

1) European distance method

$$D(d_i, d_j) = \sqrt{\frac{1}{N}(\Sigma(w_{ik} - w_{jk})^2)} \tag{17}$$

Where d_i denotes the feature vector of a certain text in the training set, d_j denotes the feature vector of another text in the training set, N is the feature vector dimension, and w_k is the kth dimension of the feature vector.

The Euclidean distance method states that the smaller the distance, the higher the correlation between the two medical health data; the greater the distance, the lower the correlation between the two medical health data. The Euclidean distance method is used to calculate the similarity. The method is simple and the calculation speed is fast, but the classification effect is often not ideal.

2) Angle cosine method

$$Sim(d_i, d_j) = \frac{\Sigma w_{ik} \times w_{jk}}{\sqrt{(\Sigma w_{ik}^2)(\Sigma w_{jk}^2)}} \tag{18}$$

The meaning of each parameter is consistent with the Euclidean distance method.

For angle cosine method, the smaller the angle between the two medical health data, the higher the degree of correlation, which means the higher the similarity; the larger the angle between the two medical health data, the lower the correlation. This means that the lower the similarity. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

similarity result calculated by the cosine method is [0,1]: when the two medical health data are completely unrelated, the similarity is 0; when the two medical health data are highly correlated, the similarity approaches to 1. Text classification systems generally use the cosine cosine method to calculate text similarity.

B. Density cropping based on cluster denoising in KNN algorithm

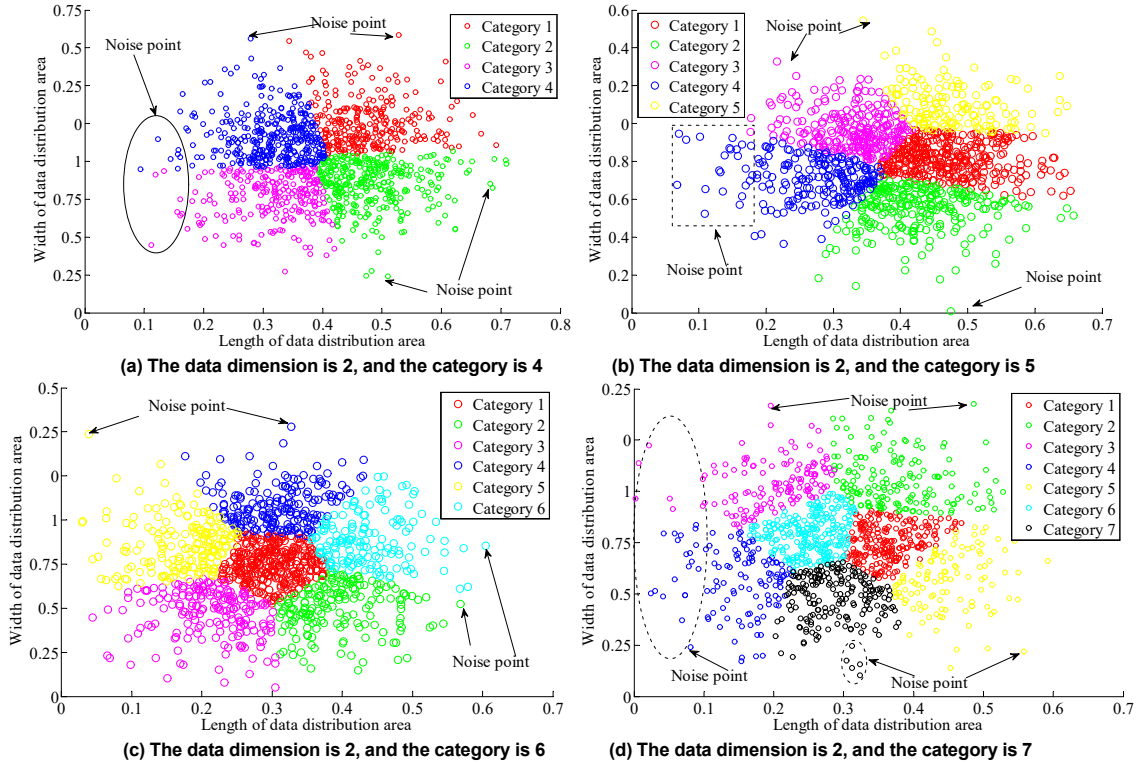
1) Basic idea of the algorithm

On the one hand, the KNN algorithm is a classification algorithm based on local information processing. It has no good grasp of global information, so the existence of noise has a certain influence on the classification accuracy of KNN algorithm. By clustering, the texts of the same category in the training set are clustered into several sub-categories according to the degree of similarity, and several cluster centers are formed. In the clustering process, those texts with lower similarity to most texts are performed. This can effectively reduce the impact of noise on the classification accuracy of the KNN algorithm.

On the other hand, the calculation time of the KNN algorithm is mainly consumed in calculating the similarity between the text to be classified and the text of the training set, which is a repeated calculation work, and the calculation time is linear with the number of training set texts. Therefore, if a part of the text with a more representative classification is selected from the training set to represent the training set, the computational complexity of the KNN algorithm will be greatly reduced, and the time efficiency of the KNN algorithm can be improved, especially the KNN algorithm is processed in the processing of large data sets.

2) Cluster denoising

Clustering is to cluster n_c texts in the same category c_i into m_c (m_c<n_c) small clusters. The similarity of texts in the same cluster is the largest, and the similarity of texts in different clusters is relatively small. After clustering, some texts that are not classified into any small cluster appear. Generally, these texts belong to noise text, and the representativeness of the category is relatively small. After removing these noise texts, the degree of classification accuracy of the classification algorithm is guaranteed.



(a) The data dimension is 2, and the category is 4

(b) The data dimension is 2, and the category is 5

(c) The data dimension is 2, and the category is 6

(d) The data dimension is 2, and the category is 7

FIGURE 7.**Removing training set noise during clustering**

It can be seen from Figure 7 that for different types of samples, there is a big difference in the degree of similarity between samples due to the differences between the samples and the ability of each sample to express different categories. By clustering, the samples in the same category are examined and labeled once, and the cluster centers are calculated and selected, and the samples with higher similarity are clustered into one small cluster to form multiple clusters. In the process, there will be a few samples that have not been added to any cluster, as shown in the noise points in Figure 7, the noise text is less similar to other texts in the same category, that is to say, the representativeness of the categories is relatively poor. When the classification is judged, it is easy to mislead the classification results, resulting in misclassification. Therefore, removing these noise texts is advantageous for improving the classification accuracy of the algorithm.

3) Density cutting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

After the clustering process, the texts of each major class in the training set are divided into several small clusters one by one, and the text density of these small clusters is relatively high. It can be seen from the specific process of the DBSCAN clustering algorithm that the text density closer to the cluster center is relatively high, and the text density at the cluster boundary is relatively low.

According to the neighbor rule of the KNN algorithm, when the test text needs to be classified, only the category attribute of the K text closest to the test text is considered. If the test text is in the central area of a class, even if the text density of the center of the class becomes smaller, the classification result of the test text will not be affected; on the contrary, if the test text is in the class boundary area of some classes, then the text of the class boundary area of several classes is examined, which is also independent of the text density of the class center area of each class. The text of the class center contributes less to the classification at this time. Therefore, if you can crop the text with very high text density in the center-like area, only retain some of the text in the center of the class, and reduce the density of the center of the class, which can greatly reduce the classification time for classifying the test set using the KNN algorithm.

Excessively high text density in the training set will affect the classification efficiency of the KNN algorithm, and will lead to misclassification to some extent. The main reason is that the KNN algorithm considers the category of K training samples closest to the test text when discriminating the test text. The use of a simple voting mechanism, which will lead to a higher text density category in voting has a greater advantage.

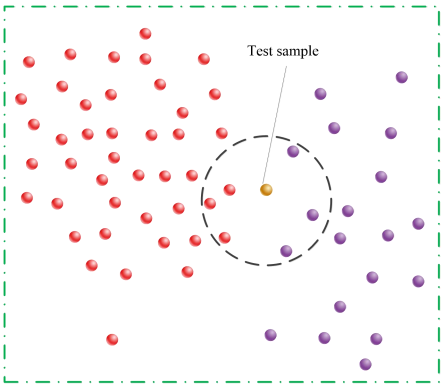


FIGURE 8.**The adverse effect of uneven text density on classification discrimination**

As shown in Figure 8, the intuitive judgment test sample should be classified into the blue category, but since the red category sample density ratio is much larger, the voting process is selected when 10 nearest neighbors are selected to discriminate the category of the test sample. The sample in the red category will have a large advantage, so the test sample will be classified into the red category and misclassified.

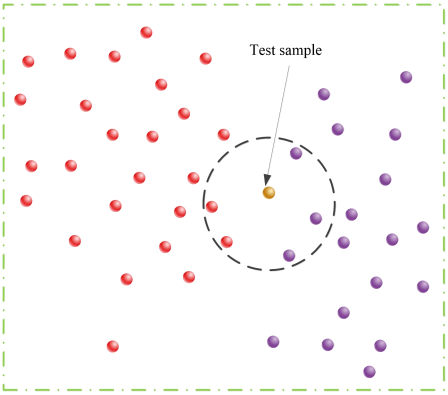


FIGURE 9.**Schematic diagram of sample classification after density cutting**

As shown in Figure 9, after density cropping, the density of the red category samples is reduced, and the problem of different voting weights due to the density difference before the cropping is solved, and the test text is correctly classified into the purple category sample set.

The main purpose of density cropping is to preserve text in areas with lower density (usually near the border of the class), and to crop the text in areas with higher density (usually near the center of the class) to reduce the amount of training text. Therefore, the number of calculations when the KNN algorithm classifies the test set is directly reduced to improve the classification efficiency.

4) Determination of main parameters

In the improved KNN algorithm based on cluster denoising and density clipping, two important parameters are the size of the neighborhood and the MinPts value. If the neighborhood selection is small and the MinPts selection is large, the small clusters of each category will increase and the text density of each cluster will be higher; if the neighborhood selection is larger and the MinPts is smaller, this will result in a reduction in the small clusters for each major class and a lower level of text intensiveness per cluster.

The determination of the neighborhood is generally based on the average neighborhood size of MinPts based on the training text set. Assuming that $L_k(t_i)$ is the distance between the text of the training text set that is closer to the kth of the specified text t_i, the average neighborhood size based on MinPts in the training text set D is calculated as follows:

$$Avg_g(D) = \frac{\Sigma L_{MinPts}(t_i)}{N} \tag{19}$$

Where N is the number of texts in the training text set D and t_i is the text in the training text set.

The determination of MinPts is generally based on the empirical value, taking 5% to 8% of the average sample size of the category, which has a good effect and has less influence on the classifier.

C. Evaluation and discussion of experimental results

1) Compare the effect of sample cutting on the performance of the classifier when K takes different values

We take 5%~8% of the average sample size of the category as MinPts value, MinPts=17. K compares the classification effect of the traditional KNN classifier with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

sample-cut KNN classifier in the medical-related text provided by a hospital at 15, 20, 25, and 30, respectively.

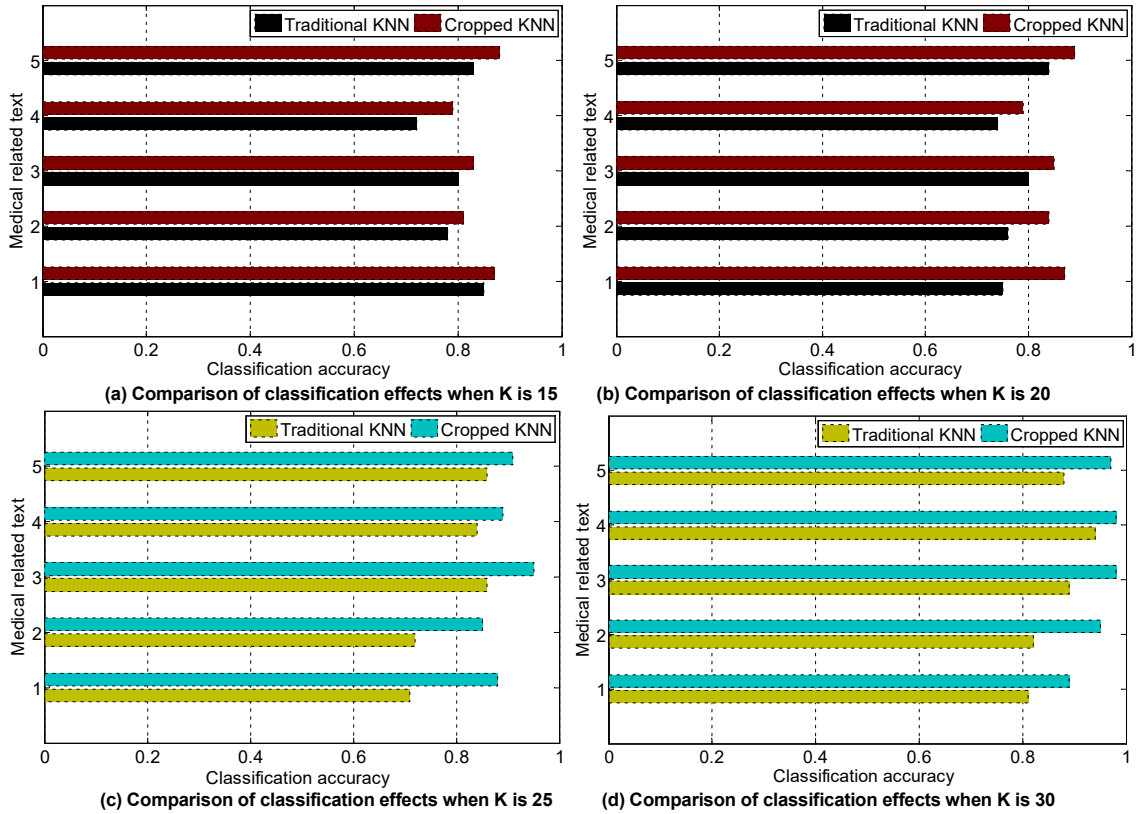


FIGURE 10. Comparison of classification effects of two KNN classifiers

It can be seen from the experimental results that when K takes different values, the classification accuracy of KNN algorithm after clustering denoising and density clipping for training set samples is improved compared with the traditional KNN algorithm except for some small categories. Under normal circumstances, there is a 1% to 2% reduction. However, due to the cropping of the sample (the cutting rate is about 20%), when the KNN algorithm classifies the test set, the calculation amount is significantly reduced, and the overall operation time is increased by about 20%. The KNN algorithm is used to classify the big data set. At the time, the improvement of time efficiency is considerable.

2) Compare the impact of the density threshold of sample clipping on the performance of the classifier

The training sample is cropped using the sample cropping algorithm, as shown in Table 2.

TABLE 2
Training sample cutting situation table

Density cropping threshold	Number of crops	Crop ratio (%)
1	1722	49.2
3	1451	41.3
5	1200	34.2
7	1069	29.9
9	981	28.0
11	932	26.7
13	845	24.2
15	801	22.7
17	735	21.2

As can be seen from Table 2, as the density threshold of the sample crop increases, the cropping ratio gradually decreases: when the density threshold is 1, the cropping rate is close to 50%; when the density threshold is equal to the MinPts value, the cropping rate is still about 20%. That is to say, the time complexity of the KNN classifier is reduced proportionally, which can greatly improve the time efficiency of the KNN classifier. However, this time efficiency is not greatly improved by sacrificing the classification accuracy. The density threshold will be compared below.

When comparing different density thresholds, the classification effect of the traditional KNN classifier and the sample-cut KNN classifier under the medical-related text provided by a hospital is shown in Figure 11.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2955754, IEEE Access

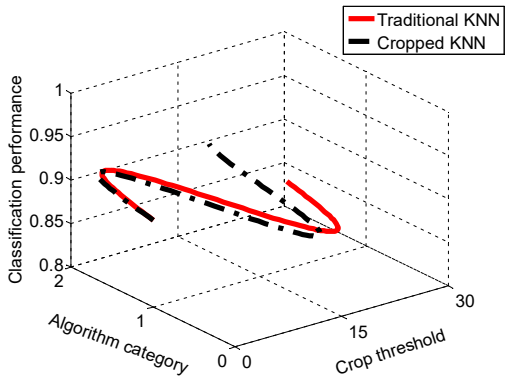


FIGURE 11. **Effect of different cropping thresholds on classification performance**

It can be seen from the experimental results that the KNN algorithm for clustering denoising and density clipping of training set samples has a great improvement in classification speed compared with the traditional KNN algorithm, and can be improved with different cutting ratios. The classification accuracy is slightly reduced due to the large-scale cropping of the training samples. It can be seen that there is better classification performance when the cropping threshold is selected between 17 and 21. Therefore, from a comprehensive perspective, the KNN algorithm for clustering denoising and density tailoring of training set samples has a certain degree of improvement in classification performance compared with the traditional KNN classifier. The loss classification accuracy can be considered to be within an acceptable range.

V. CONCLUSION

In this paper, an improved KNN algorithm is proposed to make up for some shortcomings of the traditional KNN algorithm, and the expected purpose is achieved, which reduces the running time of the KNN algorithm. A class-based weighted K nearest neighbor algorithm is proposed, which focuses on the adjustment. KNN classifier to consider the local class distribution around the query point during classification. This paper has modified the algorithm for the latest classifier on the dataset. The data sets used have different minority category percentages. The advantages and disadvantages of traditional KNN classifiers are analyzed, and an improved KNN algorithm based on cluster denoising and density cropping is proposed to overcome the shortcomings of traditional KNN algorithm in dealing with large data sets. The algorithm pre-classifies the training set by clustering and density clipping of the training set samples to speed up the K-nearest neighbor search speed and improve the classification efficiency of the KNN algorithm in processing large data sets, while maintaining the KNN algorithm. Classification accuracy is within acceptable limits. This paper studies single-class classification, but in practical applications, multi-class classification is more and more popular. The reason is that some samples have multi-category information, and they have certain ability to express two or more categories. If these samples are grouped into a fixed category according to the criteria for single-category classification, the sample sparsity of the category will increase, and the category discrimination will decrease. Research on multi-category classifications will also be the focus of future work. In addition, the medical health data also has a relatively high field missing rate, and the field missing rate has a great influence on the classification results of this paper. How to improve the medical health data through certain methods to obtain a relatively more accurate data set will be the future research direction.

REFERENCES

[1] Alimjan G, Sun T, Yi L, et al. “A New Technique for Remote Sensing Image Classification Based on Combinatorial Algorithm of SVM and KNN”, *International Journal of Pattern Recognition & Artificial Intelligence*, vol.32, no.7, pp.1859012, 2018.

[2] Lei L, Xiu-Min C, Zhong-Lian J, et al. “Ship trajectory classification algorithm based on KNN”, *Journal of Dalian Maritime University*, vol.44, no.3, pp.15-21, 2018.

[3] Chen Z. “Identification of Android Malicious Behaviors Based on k Nearest Neighbor Algorithm and Least Squares Support Vector Machine”, *Journal of Jilin University*, vol.53, no.4, pp.720-724, 2015.

[4] Zhang S, Li X, Zong M, et al. “Efficient kNN Classification With Different Numbers of Nearest Neighbors”, *IEEE Transactions on Neural Networks & Learning Systems*, vol.29, no.5, pp.1774-1785, 2018.

[5] Abbasi-Kesbi R, Nikfarjam A, Hezaveh A A. “Developed wearable miniature sensor to diagnose initial perturbations of cardiorespiratory system”, *Healthcare Technology Letters*, vol.5, no.6, pp.231-235, 2018.

[6] Zhan Y, Shan D, Mao Q, et al. “A Video Semantic Analysis Method Based on Kernel Discriminative Sparse Representation and Weighted KNN”, *Computer Journal*, vol.58, no.6, pp.1360-1372, 2018.

[7] Feng J, Wei Y, Zhu Q. “Natural neighborhood-based classification algorithm without parameter k”, *Big Data Mining & Analytics*, vol.1, no.4, pp.257-265, 2018.

[8] Hu B, Li X, Sun S, et al. “Attention Recognition in EEG-Based Affective Learning Research Using CFS+KNN Algorithm”, *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol.15, no.2, pp.38-45, 2018.

[9] Dobkin B H, Martinez C. “Wearable Sensors to Monitor, Enable Feedback, and Measure Outcomes of Activity and Practice”, *Current Neurology and Neuroscience Reports*, vol.18, no.12, pp.87, 2018.

[10] Durongbhan P, Zhao Y, Chen L, et al. “A Dementia Classification Framework Using Frequency and Time-Frequency Features Based on EEG Signals”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol.27, no.5, pp.826-835, 2019.

[11] Xu C, Wang Y, Bao X, et al. “Vehicle Classification Using an Imbalanced Dataset Based on a Single Magnetic Sensor”, *Sensors*, vol.18, no.6, pp.1690, 2018.

[12] Venkatesan C, Karthigaikumar P, Varatharajan R. “A novel LMS algorithm for ECG signal preprocessing and KNN classifier based abnormality detection”, *Multimedia Tools & Applications*, vol.77, no.2, pp.1-10, 2018.

[13] Noi P T, Kappas M. “Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery”, *Sensors*, vol.18, no.1, pp.18, 2018.

[14] Zhang R, Peng P, Dai Q, et al. “Sensitive and wearable carbon nanotubes/carbon black strain sensors with wide linear ranges for human motion monitoring”, *Journal of Materials Science Materials in Electronics*, vol.29, no.7, pp.5589-5596, 2018.

[15] Zainuddin A Z A, Mansor W, Khuan L Y, et al. “Classification of

EEG Signal from Capable Dyslexic and Normal Children Using KNN", *Advanced Science Letters*, vol.24, no.2, pp.1402-1405, 2018.

[16] Yousef M, Khalifa W, Abdallah L. "Ensemble Clustering Classification Applied to Competing SVM and One-Class Classifiers Exemplified by Plant MicroRNAs Data", *J Integr Bioinform*, vol.13, no.5, pp.11-21, 2016.

[17] Bhavani R R, Jiji G W. "Image registration for varicose ulcer classification using KNN classifier", *International Journal of Computers & Applications*, vol.40, no.4, pp.1-10, 2017.

[18] Ding J, Cheng H D, Min X, et al. "Local-weighted Citation-kNN algorithm for breast ultrasound image classification", *Optik - International Journal for Light and Electron Optics*, vol.126, no.24, pp.5188-5193, 2015.

[19] Yigit H. "ABC-based distance-weighted kNN algorithm", *Journal of Experimental & Theoretical Artificial Intelligence*, vol.27, no.2, pp.10, 2015.

[20] Al‐Ammar A S, Barnes R M. "Supervised cluster classification using the original n‐dimensional space without transformation into lower dimension", *Journal of Chemometrics*, vol.15, no.1, pp.49-67, 2015.

[21] Pohjalainen J, Räsänen O, Kadioglu S. "Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits ☆", *Computer Speech & Language*, vol.29, no.1, pp.145-171, 2015.

[22] Roy K, Ghosh S K, Sultana A, et al. "A Self-Powered Wearable Pressure Sensor and Pyroelectric Breathing Sensor Based on GO Interfaced PVDF Nanofibers", *ACS Applied Nano Materials*, vol.2, no.4, pp.2013-2025, 2019.

[23] Wang X H, Liu A, Zhang S Q. "New facial expression recognition based on FSVM and KNN", *Optik - International Journal for Light and Electron Optics*, vol.126, no.21, pp.3132-3134, 2015.

[24] Juan L I, Wang Y P. "An Incremental Learning Vector Quantization Algorithm Based on Pattern Density and Classification Error Ratio", *Acta Automatica Sinica*, vol.41, no.6, pp.1187-1200, 2015.

[25] Li X, Du Z, Wang J, et al. "In Silico Estimation of Chemical Carcinogenicity with Binary and Ternary Classification Methods", *Molecular Informatics*, vol.34, no.4, pp.228-235, 2015.

[26] Shengyun Liang, Yunkun Ning, Huiqi Li. "Feature Selection and Predictors of Falls with Foot Force Sensors Using KNN-Based Algorithms", *Sensors*, vol.15, no.11, pp.29393-29407, 2015.

[27] Nitin S, Denise J, Taylor G W, et al. "Robotic pilot study for analysing spasticity: clinical data versus healthy controls", *Journal of Neuroengineering & Rehabilitation*, vol.12, no.1, pp.109, 2015.

Wenchao Xing graduated in Computer Science and Technology in 2004 from the Shandong Normal University, Shandong, China and the MS degree in Management Science and Engineering in 2010 from the Shandong Normal University, Shandong, China. He is a lectuer in JiNing University. His research interests include image processing ,cyberspace security,internet of things, security,magagement science and engineering.e-mail: xww2001@163.com

Yilin Bei received the BS degree in Computer Science and Technology in 2003 from the Shandong Normal University, Shandong, China and the MS degree in Computer Software and Theory in 2008 from the Liaoning Normal University, Dalian, China . He is an associate professor in TaiShan University. His research interests include digital watermarking, virtual reality and information security.e-mail: beiyilinok@163.com