Research Article

An Efficient and Fast Model Reduced Kernel KNN for Human Activity Recognition

Zongying Liu, Shaoxi Li, Jiangling Hao, Jingfeng Hu, and Mingyang Pan

Dalian Maritime University, Faculty of Navigation, No. 1 Linghai Road, Dalian 116085, China

Correspondence should be addressed to Shaoxi Li; lishaoxi@dlmu.edu.cn

Received 8 April 2021; Accepted 26 May 2021; Published 3 June 2021

Academic Editor: Chunjia Han

Copyright © 2021 Zongying Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With accumulation of data and development of artificial intelligence, human activity recognition attracts lots of attention from researchers. Many classic machine learning algorithms, such as artificial neural network, feed forward neural network, K-nearest neighbors, and support vector machine, achieve good performance for detecting human activity. However, these algorithms have their own limitations and their prediction accuracy still has space to improve. In this study, we focus on K-nearest neighbors (KNN) and solve its limitations. Firstly, kernel method is employed in model KNN, which transforms the input features to be the high-dimensional features. The proposed model KNN with kernel (K-KNN) improves the accuracy of classification. Secondly, a novel reduced kernel method is proposed and used in model K-KNN, which is named as Reduced Kernel KNN (RK-KNN). It reduces the processing time and enhances the classification performance. Moreover, this study proposes an approach of defining number of K neighbors, which reduces the parameter dependency problem. Based on the experimental works, the proposed RK-KNN obtains the best performance in benchmarks and human activity datasets compared with other models. It has super classification ability in human activity recognition. The accuracy of human activity data is 91.60% for HAPT and 92.67% for Smartphone, respectively. Averagely, compared with the conventional KNN, the proposed model RK-KNN increases the accuracy by 1.82% and decreases standard deviation by 0.27. The small gap of processing time between KNN and RK-KNN in all datasets is only 1.26 seconds.

1. Introduction

Since the 21st century, the development of Internet and the popularity of wearable devices brings data explosion. These massive data provide the foundation of structuring artificial intelligence (AI) algorithms. In recent decades, machine learning algorithms as backbone of AI make great progress in many domains, for example, wind speed prediction in Brazil by machine learning algorithms [1], weather time series prediction by fuzzy models [2], automatic sleep stage classification by convolution neural network [3], and the diseases classification [4, 5].

At the same time, with development of microprocessor and enabling sensors with high computational ability, the small size, and low cost, portable devices, such as smartphone, band, smart watch, or professional sensors, are used widely and record the huge data from users. Moreover, machine learning algorithms successfully solve problems in our daily life [6–8]. Nowadays, activity recognition using wearable sensors data attracts attention from researchers and businessman. Machine learning algorithms are also applied in variant intelligent devices, such as intelligent band with activity detection element that records the different activity situation for keeping health, or iWatch with fall detection function, which assists in monitoring older people activities and alarming dangerous activities. The state-of-the-art classifiers successfully achieve in human activity recognition. Artificial neural network (ANN) is a widely used model that models the relationship between input and output units. The most popular training algorithm in ANN is the backpropagation which iterative learns a set of weights for the prediction of the class labels. Paper [9] also applies optimal ANN classifier to recognize human activity based on mobile sensors data. However, the backpropagation
algorithm takes long time for seeking the suitable weights of ANN. It limits the environment of application. The other famous model is called support vector machines (SVM). Palaniappan et al. [10] combined the widely used approach SVM with a novel scheme of representing human activities for classifying human activity. Due to kernel computation, SVM also needs to face the same problem as model ANN. Besides, $k$-nearest neighbors (KNN) is a simple, yet effective classification recognition machine learning algorithm that is widely applied. It is applied to build a human activity recognition system and obtains a significantly outstanding performance [11].

Although KNN has less computation than other classic machine learning algorithms, it still has limitations in the processing of classification. There are three main aims in this study. The first one is to enhance the classification performance comparing with the conventional KNN. This study employs kernel method to transform the input data to be high-dimensional features. Secondly, KNN will face the heavy computation when data is large scaled. Thus, the second aim is to propose a novel reduced kernel to decrease the processing time and further increase classification accuracy. Lastly, the parameter dependency problem in KNN is a hot topic. This study proposes the way of defining the number of $K$ neighbors that achieves the best classification performance than others. Therefore, four contributions of this study are briefly described as follows:

(i) The kernel method is applied in KNN for gaining the high-dimensional features that impact the classification performance in the positive way.

(ii) A novel reduced kernel approach is proposed. It is successfully applied to reduce the heavy computation of kernel method and increase the computing efficiency. An efficient and fast model called Reduced Kernel KNN is proposed.

(iii) A method of defining number of $K$ neighbors is proposed, which obtains the outstanding performance in classification compared to others.

(iv) The proposed model not only has the superior ability in human activity recognition, but also achieves the better performance in benchmarks compared with other models. Thus, it has generalized ability in classification.

The paper is organized as follows. Section 2 reviews the conventional model KNN. Section 3 describes our new methods and proposed model. Section 4 explains the experiment setup and evaluation of our algorithm against baselines and proves the modifying model has influence on the classification performance. Section 5 provides conclusion.

### 2. Preliminary Works

KNN attracts lots of attention from researchers and project developers. Due to its easy-to-implement characteristic, it is widely applied to solve both classification and regression problems, such as financial time series prediction [12], short-term traffic flow prediction [13], recognition of diseases in paddy leaves [14], and human activity recognition [11]. Although it achieves the good performances in the different domains, it still has limitation for dealing with large-scaled data. The computation complexity will increase in KNN when the size of data in use grows. Before introducing our proposed method, let us briefly represent the working process of KNN first.

In classification aspect, KNN applies feature distance to predict the coming sample how closed to the points in the training set. It categorizes the sample with the closest feature distance into the specific category. The output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its $K$-nearest neighbors. Generally, $K$ is a positive integer and small value in KNN. The working process is introduced as the following steps: firstly, we assume that $X = \{x_{1,1}, x_{2,1}, \ldots, x_{N,1}\} \in \mathbb{R}^{N \times P}$ is the input features and $i = \{1, 2, \ldots, N\}$, $j = \{1, 2, \ldots, D\}$, where $N$ represents the size of data and $P$ is the number of features. At the same time, $Y$ is a vector that represents the labels of the corresponding input features. Secondly, the value of $K$ is supposed to define by user, which directly impacts the performance of classification. Next, it calculates the distance between the training samples and the current sample from the testing data. Generally, to avoid the matching problem between objects, the distance is computed by Euclidean Distance equation. Assume that there are two objects $a$ and $b$ with N-dimension. The equation of distance is shown as follows:

$$D(a, b) = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2}. \quad (1)$$

And, set an index based on all distances and then sort them in ascending order. Then, it will choose the top $K$ rows from the sorted index. And then, keep the top $K$ rows from all classes in the sorted index. Finally, return the class label based on the most frequent class of processed index. Pseudocode for KNN is shown in Algorithm 1.

Based on previous research studies on the conventional KNN and variant KNN, many research works played a vital role in classification. For example, a multi-label lazy learning approach based on KNN, which mainly applied maximum a posteriori (MAP) principle to gain statistical information from training data, determines the label set for the unseen instance [15]. Moreover, the ensemble algorithm based on KNN also attracted attention. Xiao proposed an ensemble learning algorithm that applied support vector machine and KNN for traffic incident detection [16]. On the other hand, researches paid more attention to the problem of definition of parameter $K$ in variant KNN, which directly impacted the performance in classification. Zhang et al. [17] proposed a S-KNN algorithm to identify an optimal $K$ value. Sharma [18] applied DBSCAN to set parameter according to information of the data as it got accumulated in a cluster structure.

Therefore, according to the characteristic of KNN and variant KNN, they have two main drawbacks existing in the process of classification. The parameter of KNN and variant...
KNN impacts the performance of classification. The computation complexity will grow sharply when the size of data increases. Hence, the main aims of this study are to build an efficient and fast classification model based on KNN for solving these two problems and enhance the classification performance, especially for human activity. The following will explain the proposed methods with improvement of the classification performance and reduction of computation complexity.

3. Methodology

For the large-scaled data, the time consumption of KNN is largely researched in the machine learning. Although KNN has efficient classification, the heavy computation usually is a barrier for applying in the real-world project when the training data become large. In this section, an efficient and fast model is introduced based on the baseline model KNN for improving the classification performance in some extent and increasing the training efficiency. Firstly, data processing is described. Secondly, the model KNN with kernel method is introduced, where kernel method is applied in KNN. It employs the characteristic of kernel method for expanding the dimension of features and enhances the performance of classification in the baseline model KNN. Next, reduced kernel method is introduced. It mainly focuses on reducing the kernel computation. Finally, the detail of our proposed method is shown.

3.1. Data Processing. In this section, we assume that the dataset \( (X) \) with \( N \) samples is represented as follows:

\[
X = [x_{1j}, x_{2j}, \ldots, x_{nj}] = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} & Y_1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} & Y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D} & Y_N \end{bmatrix}, \tag{2}
\]

where \( x_{ij} \) is the element feature in the \( i \)-th row and \( j \)-th column from data matrix \( X \) and \( D \) is represented by the number of features. The last column of \( X \) is the corresponding \( i \) label \( (Y_i) \).

Then, the data \( (X) \) is separated by the training data \( (TR) \) and testing data \( (TX) \). Their corresponding features and labels are \( X_{tr}, Y_{tr} \) and \( X_{tx}, Y_{tx} \), respectively. Here, we set the number of training data as \( L \). The matrix data \( (X) \) can be processed as follows:

\[ X_{tr} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{L,1} & x_{L,2} & \cdots & x_{L,D} \end{bmatrix}, \tag{3} \]

\[ X_{tx} = \begin{bmatrix} x_{L+1,1} & x_{L+1,2} & \cdots & x_{L+1,D} \\ x_{L+2,1} & x_{L+2,2} & \cdots & x_{L+2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D} \end{bmatrix}, \tag{4} \]

where their corresponding labels are \( Y_{tr} = [y_1, y_2, \ldots, y_L] \) and \( Y_{tx} = [y_{L+1}, y_{L+2}, \ldots, y_N] \).

3.2. KNN with Kernel Method. Kernel method is applied in the different algorithms, such as extreme learning machine [19], support vector machine [20], online learning algorithms [21, 22], and multilayer algorithms [23, 24], which obtain the good performance in regression and classification. The kernel method transforms the low-dimensional features to the high-dimensional features. In this study, we employ Gaussian Kernel method \( (G(\ldots)) \) to process the features. Its mathematical equation is shown in the following equation:

\[
G(A, B, \sigma) = e^{-\left((A-B)^T(A-B)\right)/2\sigma^2}, \tag{5}
\]
where $\mu_A$ and $\mu_B$ are the mean of input data $A$ and $B$, respectively, and $\sigma$ represents the kernel parameter that is defined by user.

This study employs Gaussian kernel method to provide high-dimensional features for the classification datasets. It is used to combine KNN algorithm in order to improve its classification ability. This algorithm is proposed and named as k-nearest neighbor with kernel (K-KNN). The main working process of K-KNN is separated by the four parts. Firstly, the features of data, including $X_{tr}$ and $X_{tx}$, are transformed by Gaussian kernel method. The kernel matrix of training features ($KX_{tr}$) and testing features ($KX_{tx}$) is computed by equations (6) and (7), respectively:

$$KX_{tr} = G(X_{tr}, X_{tr}, \sigma),$$ (6)

$$KX_{tx} = G(X_{tr}, X_{tx}, \sigma).$$ (7)

Secondly, the distances between all testing features and training feature matrix are calculated by equation (1). Next, sort the distance for each class in ascending order and then pick the top $K$ elements from the sorted collection index. Finally, the prediction class will be voted based on the most frequent class of processed index. The main process of K-KNN is shown in Algorithm 2.

However, the Gaussian kernel method brings the heavy computation with the increasing scale of data, which directly leads to the decrease of efficiency in model K-KNN. To solve this limitation and enhance the efficiency of computation in the process of classification, the reduced kernel method is proposed in this study. The following section will introduce the details.

3.3. Reduced KNN with Kernel Method. This section proposes a new model named Reduced KNN with kernel method (RK-KNN). It applies the reduced kernel method that replaces the intrinsic kernel method in the model K-KNN.

The main idea of reduced kernel method is to apply a certain percentage of training data from each class to calculate the kernel matrix, which directly reduces the computation complexity of kernel method. Generally, variant models with kernel method have the good performance in regression and classification. For instance, extreme learning machine with kernel (KELM) [25, 26] even obtains better performance in regression and classification than the conventional extreme learning machine. The classic model support vector machine (SVM) applies kernel method to connect the input layer with the hidden layer, which provides the high-dimensional support vectors to transform the original input features. The kernel method of the model SVM also plays a vital role in enhancing the classification performance. These algorithms use the entire input to compute the kernel matrix. This study claims that the kernel matrix is computed by the randomly selected samples from the training observations. This type of features representation is the same or even better than that of the conventional kernel matrix. The mathematical equation of reduced kernel is represented as follows:

$$KM = G(A_P, B, \sigma),$$

where $KM$ is the reduced kernel matrix, $P$ represents the selected percentage for computing kernel matrix, and $A_P$ stands for the selected $P$ samples from input data $A$.

Then, the reduced kernel matrix of training and testing data can be calculated by equations (9) and (10), respectively:

$$KX_{tr} = G(X_{trP}, X_{tr}, \sigma),$$ (9)

$$KX_{tx} = G(X_{trP}, X_{tx}, \sigma),$$ (10)

where $X_{trP}$ represents the $P$ percentage samples from each class in the training observations $X_{tr}$.

The kernel matrix of K-KNN can be replaced by the reduced kernel matrix, which has less computation of kernel matrix than full kernel matrix in model K-KNN. Reduced kernel method not only keeps the high-dimensional features from kernel method, but also reduces the computing process of kernel matrix by selecting certain percentage of training samples. Therefore, the brief pseudocode of RK-KNN is shown in Algorithm 3.

4. Experimental Works

To prove that the kernel method and reduced method play a vital role in the enhancement of classification performance and decrease of processing time in model KNN, this section applies ten benchmarks dataset (binary and multiclass) and two real-world human activity datasets. Moreover, the parameter dependency exists in the variant models of KNN. The last part of this section indicates how to define the range of $K$ parameter in our proposed model.

4.1. Data Description and Parameter Setting. Run ten benchmarks and two human activity datasets in the experiment, which include binary and multiclass.

For the real-world data, this study uses two human activity datasets in the following experiments. The first data is Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set (HAPT) [27]. It captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50 Hz using the embedded accelerometer and gyroscope of the device. Figure 1 shows the condition of activities, including stand-to-sit, sit-to-stand, sit-to-lying, lying-to-sit, stand-to-lying, and lie-to-stand.

The other human activity data is Smartphone Dataset for Human Activity Recognition in Ambient Assisted Living Data (Smartphone) [28]. Figure 2 shows the different human activities, including standing, sitting, laying, walking, walking upstairs, and walking downstairs.

The obtained datasets are randomly partitioned into two sets, where 70% of the volunteers are selected for generating the training data and the remaining volunteers belong to the test data.

Besides, all benchmark datasets can be found in the part of classification from UCI Machine Learning Repository. The number of features and classes in benchmark and real-world datasets is shown in Table 1.
Besides, the parameter setting directly affects the performance of classification. Fairly comparison among models is necessary. Firstly, KNN and all variant KNN models need to define the number of neighbors (K). The following experiments set the number of neighbors (K) as the class size for all compared models. Secondly, due to the kernel method that is used in model K-KNN and RK-KNN, the kernel parameter is set as the same number. Lastly, to reduce kernel computation, the selected percentage is defined as 10% for benchmarks and 30% for real-world data in model RK-KNN.

4.2. Experimental Results and Discussion. Based on the parameters setting, three models are compared, including baseline model KNN, KNN with kernel method, and proposed model RK-KNN. Due to random selection for training and testing samples from all datasets, all experiments are run ten times in order to keep the generalization of classification performance. Moreover, to compare training time fairly, we run all experiments by python3.6 version under Windows 10 system with 16 GB memory and Intel 8th Generation i7 processor.
Table 2 shows the classification performances in benchmarks and real-world datasets. The accuracy is used to determine the ability level of classification. Standard deviation (SD) shows the difference among ten times prediction results. The processing time is shown in the last column (Time) of each model, which is recorded in seconds. Based on the comparison between the baseline model KNN and other two models, it demonstrates that the kernel method and the reduced kernel method play a vital role in aspect of enhancement of classification and processing efficiency. Firstly, comparing the performance of model KNN with that of K-KNN, model K-KNN has better performance in all datasets than that of KNN. Averagely, the accuracy of model K-KNN increases by 0.06% compared with the baseline model KNN. The maximum increase (1.17%) appears in Smartphone dataset. At the same time, the performance of K-KNN in HAPT data keeps the constant with that of model KNN. In aspect of standard deviation, the real-world datasets in the model K-KNN have lower values than those in the model KNN. There are similar differences in SD for benchmarks. However, the processing time in model K-KNN is almost ten times that of model KNN.

To keep the advantage of kernel method for enhancing classification ability and overcome the limitation of heavy kernel computation, we propose an efficient and fast model RK-KNN. The second experiment focuses on the role that

### Table 1: Details of datasets.

| Benchmarks        | Observations | Features | Classes |
|-------------------|--------------|----------|---------|
| Iris              | 150          | 4        | 3       |
| Heart             | 270          | 13       | 2       |
| Breast            | 263          | 9        | 2       |
| Diabetes          | 768          | 8        | 2       |
| Seed              | 210          | 7        | 3       |
| Thyroid           | 215          | 5        | 2       |
| Banana            | 5300         | 2        | 2       |
| Image             | 2086         | 18       | 2       |
| Ringnorm          | 7400         | 20       | 2       |
| Splice            | 2991         | 60       | 2       |

| Human activity data | Observations | Features | Classes |
|---------------------|--------------|----------|---------|
| HAPT                | 10926        | 561      | 6       |
| Smartphone          | 5744         | 561      | 6       |
Reducing kernel method plays in model K-KNN. The performances are compared between models K-KNN and RK-KNN. Averagely, the accuracy of model RK-KNN increases by 1.46% compared with model K-KNN. Moreover, the model RK-KNN has the best performance in all datasets. The maximum increase rate appears in Heart data, which increases 3.09% compared with model K-KNN. The minimum increase (0.20%) exists in Splice. For the real-world datasets, model RK-KNN not only develops the classification accuracy, which increased by 0.58% for HAPT and 0.29% for Smartphone, but also reduces at least ten times processing time than model K-KNN. In aspect of standard deviation, averagely, model RK-KNN has the lowest standard deviation among these three compared models. It demonstrates that model RK-KNN has more stable classification ability than others. On the other hand, the average processing time of model RK-ELM in all datasets is close to that of model KNN. Especially for the real-world datasets, the processing time of model RK-KNN is less than that of model KNN. Therefore, the proposed model RK-KNN not only enhances the classification ability in the benchmarks and human activity datasets, but also reduces the kernel computation.

4.3. Statistical Analysis. In this section, it applies Friedman test to check whether the different modifying models impact the performances from all benchmarks and real-world datasets. The Friedman testing data is a group of twelve different datasets with their classification accuracy performances on different data as a result of a change in the different modifying model. The testing data are shown in Table 3. The \( H_0 \) is that the classification accuracy will be the same regardless of the modifying model. We run Friedman test in IBM SPSS Statistics 21. Based on the testing result, in the 95% significant level, the \( p \) value of Friedman test is 0. Because \( p \) value is less than 0.05, we can reject \( H_0 \). Therefore, there is sufficient evidence that indicates the modifying model significantly altered the classification performance.

4.4. Impact of \( K \) Neighbors. \( K \) value in KNN and its variant models affects the classification performance. \( K \) value equal to one means that it will take one nearest neighbor and classify the query point based on its label. If value of \( K \) is extremely large, the model will underfit. Therefore, \( K \) value is the key element that directly impacts the classification result. To further analyze the rule of \( K \) neighbors for impact of classification performance and prove our statement regarding the definition of \( K \) neighbors in our proposed model, this section observes the different performances in a certain range of \( K \) neighbors of model RK-KNN.

### Table 2: The comparison results among three models.

| Data  | KNN      | K-KNN    | RK-KNN    |
|-------|----------|----------|-----------|
|       | Accuracy (%) | SD | Time (s) | Accuracy (%) | SD | Time (s) | Accuracy (%) | SD | Time (s) |
| Iris  | 92.66    | 2.56    | 0.03     | 93.12    | 4.05    | 0.27     | 95.20    | 2.73 | 0.05    |
| Heart | 88.51    | 3.13    | 0.21     | 88.57    | 3.49    | 1.61     | 91.66    | 2.56 | 0.27    |
| Breast| 81.49    | 4.34    | 0.19     | 82.04    | 4.36    | 1.49     | 82.77    | 4.37 | 0.49    |
| Diabetes | 83.64   | 2.34    | 2.62     | 83.85    | 2.46    | 44.19   | 84.32    | 2.23 | 11.67   |
| Seed  | 95.21    | 2.39    | 0.09     | 95.53    | 3.02    | 0.73     | 97.36    | 1.37 | 0.14    |
| Thyroid | 97.41 | 2.20    | 0.08     | 97.48    | 2.09    | 0.79     | 97.67    | 1.98 | 0.31    |
| Banana | 93.15    | 0.67    | 31.67    | 94.14    | 0.63    | 177.98   | 94.67    | 0.61 | 21.49   |
| Image | 92.09    | 1.25    | 407.32   | 92.11    | 2.25    | 2396.5   | 99.04    | 0.97 | 422.5   |
| Ringnorm | 92.14 | 1.15    | 411.79   | 92.23    | 0.61    | 2221.83  | 92.83    | 0.98 | 521.13  |
| Splice | 92.95    | 0.77    | 413.50   | 93.31    | 0.66    | 2255.67  | 93.51    | 0.65 | 350.21  |
| HAPT  | 91.02    | 0.67    | 500.43   | 91.02    | 0.58    | 4334.26  | 91.60    | 0.32 | 495.78  |
| Smartphone | 91.21 | 1.05    | 170.24   | 92.38    | 0.38    | 2325.65  | 92.67    | 0.49 | 129.20  |
| Average | 90.96    | 1.88    | 161.51   | 91.32    | 2.05    | 1146.75  | 92.78    | 1.61 | 162.77  |

### Table 3: The data of Friedman test.

| Data  | KNN      | K-KNN    | RK-KNN    |
|-------|----------|----------|-----------|
| Iris  | 92.66    | 93.12    | 95.20    |
| Heart | 88.51    | 88.57    | 91.66    |
| Breast| 81.49    | 82.04    | 82.77    |
| Diabetes | 83.64   | 83.85    | 84.32    |
| Seed  | 95.21    | 95.53    | 97.36    |
| Thyroid | 97.41 | 97.48    | 97.67    |
| Banana | 93.15    | 94.14    | 94.67    |
| Image | 92.09    | 92.11    | 99.04    |
| Ringnorm | 92.14 | 92.23    | 92.83    |
| Splice | 92.95    | 93.31    | 93.51    |
| HAPT  | 91.02    | 91.02    | 91.60    |
| Smartphone | 91.21 | 92.38    | 92.67    |
Benchmarks provide the range from two to eight for \( K \) value of neighbors. The real-world dataset are in the range between three and nine for \( K \) value of neighbors. Table 4 shows the performances of the proposed model RK-KNN in the different ranges for benchmarks and real-world data. It demonstrates that the rule of \( K \) neighbor selection and the bold number is the best performance for each data in the different range of \( K \) neighbors. The result with star indicates that the accuracy is computed by the proposed model that set the number neighbors as the class size. Based on the performance in the different number of \( K \) neighbors in model RK-KNN, the majority of datasets with the class size as neighbors appear the best performance. There are two binary class datasets, including Diabetes and Splice existing the best performance in the situation of setting number of neighbors as eight. However, the second highest accuracy is from the proposed number of neighbors setting. Besides, there is small difference between the best and second accuracy, including 0.35% for Diabetes and 1.95%, respectively. Therefore, our proposed method of setting class size as neighbors in the model RK-KNN obtains the best performance in classification for the benchmarks and real-world datasets. It can be used for defining the number of neighbors in our proposed model, which directly solves the problem of parameter dependency problem.

5. Conclusions

In this paper, an efficient and novel model named as Reduced Kernel K-Nearest Neighbors is proposed. This study mainly proposes three approaches to modify model KNN and enhance the classification ability. Firstly, the kernel method is applied to transform the input data to be high-dimensional features, which directly influences the classification performance in the positive way. It is combined with model KNN and named as K-KNN. Comparing the performance of KNN with that of K-KNN, K-KNN obtained much better performance in all datasets. Secondly, to reduce the heavy computation of kernel method, a novel approach is proposed and applied in K-KNN. It is short for RK-ELM. The main objectives of proposed model RK-KNN are to increase the efficiency and classification accuracy. The last approach is method for selection correct \( K \) parameter. Our approach is easy to seek the suitable \( K \) parameter in the proposed model. Based on the experimental works, our proposed model obtains the best performance in benchmarks and human activity data. Not only does model RK-KNN have more stable classification ability than others, but its average processing time is close to that of the conventional KNN. It reduces approximately 10 times processing time compared to the model K-KNN. Moreover, the distribution of \( K \) neighbors also proved that the proposed approach is easier to set the number of neighbors in model RK-KNN, which also obtains the highest accuracy in classification. Therefore, our proposed model has super ability in human activity recognition and plays a significant role in solving the general classification ability. In the future, we will focus on the reduced percentage in our proposed reduced method, which in some extent impacts the classification performance.

Data Availability

The benchmarks and real-world data used to support the findings of this study have been deposited in the UCI Machine Learning Repository.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

All authors have participated in conception and design, or analysis and interpretation of the data, drafted the article and revised it critically for important intellectual content, and approved the final version.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant.
References

[1] K. Ali, L. Machado, and R. O. Nunes, “Time-series prediction of wind speed using machine learning algorithms: a case study osorio wind farm, Brazil,” Applied Energy, vol. 224, pp. 550–566, 2018.

[2] E. Soares, P. Costa Jr, B. Costa, and D. Leite, “Ensemble of evolving data clouds and fuzzy models for weather time series prediction,” Applied Soft Computing, vol. 64, pp. 445–453, 2018.

[3] H. Phan, F. Andreottti, N. Cooray, O. Y. Chén, and M. De Vos, “Joint classification and prediction cnn framework for automatic sleep stage classification,” IEEE Transactions on Biomedical Engineering, vol. 66, no. 5, pp. 1285–1296, 2018.

[4] S. K. Lakshmanaprabu, K. S. Nandan Mohanty, N. Arunkumar, and G. Ramirez, “Optimal deep learning model for classification of lung cancer on ct images,” Future Generation Computer Systems, vol. 92, pp. 374–382, 2019.

[5] S. Khan, N. Islam, Z. Jan, I. Ud Din, and J. Rodrigues, “A novel deep learning based framework for the detection and classification of breast cancer using transfer learning,” Pattern Recognition Letters, vol. 125, pp. 1–6, 2019.

[6] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, “Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition,” in Proceedings of the 2016 Acm Sigmac Conference on Computer and Communications Security, pp. 1528–1540, Vienna, Austria, October 2016.

[7] S. Mohan, C. Thirumalai, and G. Srivastava, “Effective heart disease prediction using hybrid machine learning techniques,” IEEE Access, vol. 7, pp. 81542–81554, 2019.

[8] B. M. Henrique, V. A. Sobreiro, and H. Kimura, “Literature review: machine learning techniques applied to financial market prediction,” Expert Systems with Applications, vol. 124, pp. 226–251, 2019.

[9] H. Kishor, V. Rajiv, and M. Vilas, “PCA based optimal ann classifiers for human activity recognition using mobile sensors data,” in Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems, vol. 1, pp. 429–436, Springer, Berlin, Germany, May 2016.

[10] A. Palaniyappan, R. Bhargavi, and V. Vaidehi, “Abnormal human activity recognition using svm based approach,” in Proceedings of 2012 International conference on recent trends in information technology, IEEE, Chennai, Tamil Nadu, India, April 2012.

[11] S. Sani, N. Wiratunga, S. Massie, and K. Cooper, “Knn sampling for personalised human activity recognition,” Case-Based Reasoning Research and Development, vol. 99, pp. 330–344, 2017.

[12] L. Tang, H. Pan, and Y. Yao, “Pank-a financial time series prediction model integrating principal component analysis, affinity propagation clustering and nested k-nearest neighbor regression,” Journal of Interdisciplinary Mathematics, vol. 21, no. 3, pp. 717–728, 2018.

[13] L. Zhao, W. Du, D.-mei Yan, C. Gan, and J.-hua Guo, “Short-term traffic flow forecasting based on combination of k-nearest neighbor and support vector regression,” Journal of Highway and Transportation Research and Development (English Edition), vol. 12, no. 1, pp. 89–96, 2018.

[14] M. Suresha, K. N. Shreekanth, and B. V. Thirumalesh, “Recognition of diseases in paddy leaves using knn classifier,” in Proceedings of 2017 2nd International Conference for Convergence in Technology (I2CCT), pp. 663–666, IEEE, Mumbai, India, April 2017.

[15] M.-L. Zhang, Z. Mi-knn, and Z.-H. Zhou, “ML-KNN: a lazy learning approach to multi-label learning,” Pattern Recognition, vol. 40, no. 7, pp. 2038–2048, 2007.

[16] J. Xiao, “Svm and kkn ensemble learning for traffic incident detection,” Physica A: Statistical Mechanics and Its Applications, vol. 517, pp. 29–35, 2019.

[17] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, “A novel knn algorithm with data-driven K parameter computation,” Pattern Recognition Letters, vol. 109, pp. 44–54, 2018.

[18] A. Sharma and A. Sharma, “Knn-dbscan: using k-nearest neighbor information for parameter-free density based clustering,” in Proceedings of 2017 International Conference on Intelligent Computing. Instrumentation and Control Techniques (ICICICT), pp. 787–792, IEEE, Kannur, Kerala, India, June 2017.

[19] G.-B. Huang, Q.-yu Zhu, and C.-k. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” in Proceedings of 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), vol. 2, pp. 985–990, Budapest, Hungary, July 2004.

[20] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[21] Z. Liu, C. K. Loo, K. Pasupa, and M. Seera, “Meta-cognitive recurrent kernel online sequential extreme learning machine with kernel adaptive filter for concept drift handling,” Engineering Applications of Artificial Intelligence, vol. 88, Article ID 103327, 2020.

[22] Simone Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, “Online sequential extreme learning machine with kernels,” IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 9, pp. 2214–2220, 2014.

[23] Z. Liu, K. Chu, and K. Pasupa, “A novel error-output recurrent two-layer extreme learning machine for multi-step time series prediction,” Sustainable Cities and Society, vol. 10, Article ID 102613, 2020.

[24] C. M. Wong, C. Man Vong, K. Pak Wong, and J. Cao, “Kernel-based multilayer extreme learning machines for representation learning,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 3, pp. 757–762, 2016.

[25] Z. Liu and K. Chu, N. Masuyama and K. Pasupa, “Recurrent kernel extreme reservoir machine for time series prediction,” IEEE Access, vol. 6, pp. 19583–19596, 2018.

[26] A. Iosifidis, A. Tefas, and I. Pitas, “On the kernel extreme learning approach to multi-label learning,” Pattern Recognition, vol. 54, pp. 11–17, 2015.

[27] J.-l. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, “Transition-aware human activity recognition using smartphones,” Neurocomputing, vol. 171, pp. 754–767, 2016.

[28] D. Dua and C. Graff, “UCI machine learning repository,” 2017.