https://doi.org/10.31449/inf.v49i16.9600 Informatica **49** (2025) 331–350 **331**

Real-Time Motion Recognition in Special Training Systems Based on the Optimized BBO-KNN Method of Motion Morphology

Yin Xu
School of Physical Education, Henan Kaifeng College of Science Technology and Communication, Kaifeng 475001, China
E-mail: xumeili2025@163.com

Keywords: KNN dynamic weight, sports, special training

Received: June 6, 2025

The traditional sports training boxing system has problems with insufficient accuracy and poor real-time performance in high similarity action classification, and lacks adaptability to individual action differences. This article constructs a sports training system based on dynamic weight optimization KNN (BBO-KNN), aiming to improve the accuracy and real-time performance of complex action recognition, and provide technical support for personalized training. In response to the problems of insufficient accuracy (high FP rate), poor real-time performance (delay>1s), and lack of individual adaptability in high similarity action classification of traditional sports training systems, this study proposes a KNN model based on dynamic weight optimization (BBO-KNN). The model performance is optimized by fusing proprietary datasets with public datasets and using 5-fold cross validation (training/testing ratio 7:3). The experimental results validate that BBO-KNN significantly outperforms benchmark models such as LSTM (94.50%) and SVM (89.30%) in accuracy (96.20% ± 0.3%). The system performs highly similar actions such as running ↔ The FP rate of jumping has decreased to 1.6%, and the global FP rate is 1.39%.and robustness (noise interference fluctuation ± 1.2%). The classification error distribution shows its stability advantage, and the confusion matrix highlights the accurate recognition of highly similar actions (such as running → jumping). Research has shown that the BBO-KNN model effectively solves the real-time and robustness problems of motion recognition through dynamic weight optimization. In the future, it can be extended to complex movements such as gymnastics by combining visual data and adapting to individual style differences through incremental learning.

Povzetek: Članek predstavi sistem za športno vadbo, ki uporablja dinamično uteženi BBO-KNN za boljše prepoznavanje gibov.

1 Introduction

Sports special training is undergoing a profound change from traditional experience-oriented to data-driven. This transformation process presents multi-dimensional technical characteristics and systematic development bottlenecks. From a macro perspective, it can be seen that the digital penetration of modern sports training systems has reached a considerable scale. According to authoritative data from the General Administration of Sport of China in 2024, more than three-quarters of professional sports teams have deployed various wearable devices for training data collection. This proportion has nearly doubled compared to five years ago, indicating that a fundamental paradigm shift is taking place in sports training methodology.

There is a sharp intergenerational gap between the rapid popularization of hardware and the intelligence level of software systems. The widespread deployment of data acquisition equipment has not simultaneously brought about a significant improvement in training efficiency, but has exposed structural defects in data processing capabilities. The specific deficiency is the core contradiction of insufficient data utilization. Currently, only 42% of sports teams have established a complete analysis system, which means that more than half of the training data is dormant and cannot be converted into effective training decision-making basis.

The deficiency of this data value mining stems from multiple technical obstacles, including but not limited to imperfect feature engineering construction, inefficient data cleaning process, and insufficient adaptability of analysis model. What is more prominent is the static phenomenon of evaluation indicators. Up to 91% of training systems still adopt the fixed weight scoring mechanism [1]. This rigid evaluation system can't adapt to the dynamic changes of athletes' physiological parameters, resulting in systematic deviation between training programs and actual needs. In addition, the feedback delay problem further amplifies this mismatch. The decision lag of 2.3 training cycles on average makes the training adjustment always lag behind the actual state change of athletes, resulting in the time loss of training effect [2].

By deeply analyzing the technical essence behind these phenomena, we can find that the fundamental reason for the homogeneity of training programs lies in the uniformity of feature extraction dimensions and the lack of personalized modeling, which reflects the fundamental contradiction between the traditional batch computing model and the real-time decision-making needs [3]. Therefore, solving these systemic defects requires the introduction of innovative algorithm architectures and technical paradigms. There are still two key optimization spaces in current technology. The first is the balance between computing resource consumption and real-time requirements, especially the control of computational complexity when processing high-dimensional features. The second is the model generalization ability in small sample scenarios and the adaptive performance when facing new athletes or rare training situations [4].

The core innovation of KNN dynamic weight optimization technology lies in building a four-dimensional optimization space, realizing minute-level weight updates in the time dimension, and compressing the data processing delay to 1/60 of the traditional method through the sliding time window mechanism and incremental learning algorithm. Moreover, it completes multimodal data fusion in the feature dimension, integrating multi-source information such as biomechanics, physiology and biochemistry, and environmental parameters [5, 6]. In the individual dimension, it establishes an athlete-specific model and achieves efficient matching of similar samples through dynamic neighborhood search technology. In the environmental dimension, it integrates venue equipment parameters to build a complete training situation perception system. This multi-dimensional optimization architecture enables the system to process nonlinear and non-stationary training data, effectively solving the response hysteresis problem of traditional systems [7].

The traditional sports training classification system encounters issues of insufficient accuracy and poor real-time performance in high-similarity action classification, and lacks adaptability to individual motion variations. Therefore, this paper constructs a sports training system based on Biogeography-Based OptimizationKNN (BBO-KNN), aiming to improve the accuracy and real-time performance of complex action recognition and provide technical support for personalized training.

This study aimed to investigate the performance limits of the BBO-KNN (BBO optimized KNN) algorithm for high-similarity action recognition by addressing a specific research question. The specific hypothesis is whether BBO-KNN could reduce the false positive rate (FP rate) to below 2%, while maintaining a stable end-to-end processing latency below 20ms and a classification accuracy better than 95%. This goal is directly aimed at the core defects of traditional systems (such as LSTM and SVM) in high similarity action (such as running and jumping) classification, with FP rate>4.2% and delay>200ms. To achieve this assumption, the system uses the BBO algorithm to optimize the feature weight vector to enhance local feature sensitivity, combines K-Means clustering to compress the dataset size, and designs a lightweight edge architecture for real-time processing.

The implementation of minute level weight updates through sliding time windows and incremental learning relies on a triple mechanism:

(1) The 200ms sensor window slides in 10ms steps to ensure real-time feature extraction;

(2) Incremental learning only updates cluster centers (non feature weights), and adjusts secondary cluster points every 5 days through new data (as mentioned in the conclusion);

(3) The feature weight WK3 remains static, and its “dynamic” effect comes from the weight distribution optimized by BBO, while window sliding allows the model to continuously capture temporal features.

2 Related work

2.1 Research status of sports special training system

Rodriguez et al. [8] developed a multi-sensor fusion wearable system. It integrates IMU, sEMG and heart rate monitoring modules, increasing the data collection dimension to 23 physiological indicators, but there is a 15% sensor signal interference problem. The 4D optical capture solution proposed by Cizmic et al. [9] improves the motion analysis accuracy to 0.3mm, but the system construction cost is as high as 2 million yuan, making it difficult to popularize and apply. At present, non-contact monitoring technology based on millimeter wave radar can realize micro-motion capture within a range of 5m, but the sampling rate is limited to 120Hz.

The BP neural network evaluation model constructed by Balkhi et al. [10] improves the accuracy of technical action scoring to 89% in sports events, but requires more than 800 hours of labeled data training. Calderón-Díaz et al. [11] introduced the transfer learning method, which can realize personalized modeling of new athletes with only 200 samples, but the cross-event transfer error still reaches 28%. It is worth noting that the digital twin evaluation system developed by Iduh et al. [12] controls the sports action prediction error within 1.2 ° through real-time physical simulation, but it needs to be equipped with the support of a supercomputing center.

From the above research on sports special training system, the current system generally faces three major challenges: (1) the asynchronous problem of multi-source data leads to 27% information loss; (2) The lack of interpretability of the model leads to the trust crisis of coaches; (3) The contradiction between hardware portability and accuracy is prominent. It is particularly noteworthy that 82% of commercial systems still use static evaluation algorithms, which cannot adapt to the dynamic changes of athletes' status.

2.2 Application of optimization algorithm in training system

Chen et al. [13] introduced genetic algorithm into sports special training cycle planning, and improved the matching degree of training scheme by 31% through adaptive cross-mutation strategy, but there is the problem of slow iterative convergence speed (an average of 14 hours). The improved particle swarm optimization algorithm by Taborri et al. [14] optimizes the load distribution of strength training in sports events, which increases the maximum strength growth rate of athletes by 22%, but the sensitivity of the algorithm parameters is high and needs repeated debugging.

The LSTM-ATT hybrid model developed by Hanif et al. [15] achieves 92% accuracy in the evaluation of sports-specific actions, but the model needs 150,000 pieces of labeled data for training. AshokKumar et al. [16] applied reinforcement learning to optimize sports-specific strategies, which increased the athlete's scoring rate by 29%, but there was a problem of high training costs (200 hours of simulated adversarial data was required). The meta-learning method can shorten the model adaptation cycle for new athletes from 14 days to 5 days, but it has a huge demand for computing resources, requiring 4 A100 graphics cards.

The Pareto frontier algorithm proposed by Kumar et al. [17] balances technical improvement and injury risk in sports training, which optimizes the training benefit-risk ratio by 37%, but the complexity of the algorithm leads to a decrease in real-time performance and a delay of up to 11 minutes. The NSGA-III algorithm developed by Molavian et al. [18] realizes the multi-objective optimization of sports specialties and improves the competition performance by 0.8%, but it needs the support of accurate biomechanical modeling. Malamatinos et al. [19] applied fuzzy logic to optimize sports posture, and the completion of movements was increased by 19%, but the construction of rule base relies on a large amount of expert knowledge.

From the above research, the current research mainly faces three bottlenecks: (1) the contradiction between the real-time performance and accuracy of the algorithm, and the optimal system still has a delay of 5-8 minutes; (2) The model is not explainable enough, and 68% of AI decisions cannot provide reasonable explanations; (3) The cross-project transfer capability is weak, and the average error is 37%. In particular, 82% of commercial training systems (2024 market research) still adopt static optimization strategies, which are difficult to adapt to the dynamic changes of athletes' status.

2.3 Research status of application of KNN optimization algorithm in sports training

The weighted dynamic KNN model proposed by Merzah et al. [20] improves the accuracy rate to 94% in sports action recognition, but the real-time calculation delay is still 1.2 seconds. The quantized distance calculation method developed by Bunker et al. [21] can increase the speed of athletes' posture analysis by 100 times, but it needs the support of special quantum computing equipment. The improved KNN scheme combined with SHAP value interpretation proposed by Teixeira et al. [22] reduces the error of sports action evaluation from 3.2° to 0.8°, but the complexity of feature engineering increases by 3 times.

The sliding window incremental learning system applied by Woltmann et al. [23] shortens the update cycle of the training model to 15 minutes, but the memory footprint is still as high as 32GB. The multi-modal distance measurement method proposed by Sonalcan et al. [24] combines electromyographic and mechanical characteristics in sports events, so that the prediction error of action angle is < 0.5 °, but 17 sensor data need to be synchronized.

Table 1 below summarizes the relevant work:

Table 1: Summary of related work

Technical direction	Method and Key Results	Limitations and technical bottlenecks
Multi sensor fusion system	Method: Multi sensor wearable system (IMU/sEMG/heart rate) Results: 23 physiological indicators were collected, and the data dimension was increased by 300%	Limitations: 15% signal interference Bottleneck: Asynchronous multi-source data leads to 27% information loss, and there is a contradiction between portability and accuracy (the cost of a 0.3mm precision system reaches 2 million yuan)
4D optical capture scheme	Method: High precision optical marker point tracking Result: Motion capture accuracy reaches 0.3mm	Limitations: Supercomputing Center Dependency (2-million-yuan cost) Bottleneck: High hardware deployment costs, difficult to popularize applications

BP neural network model	Method: Multi layer backpropagation network Result: Technical action scoring accuracy rate of 89%	Limitations: 800 hours of annotated data training required Bottleneck: high data dependency, long model update cycle (>2 weeks)
Transfer learning program	Method: Cross athlete feature transfer Result: New athlete modeling only requires 200 samples	Limitations: Cross project migration error of 28% Bottleneck: weak domain adaptability, insufficient generalization ability
Dynamic KNN algorithm	Method: Weighted Neighbor Classification Result: Action recognition accuracy rate of 94%	Limitations: Real time latency of 1.2 seconds Bottleneck: Low computational efficiency, unable to meet real-time requirements of<100ms
Quantum distance calculation	Method: Quantization feature similarity measurement Result: Attitude analysis speed increased by 100 times	Limitations: Requires specialized quantum devices. Bottleneck: Strong hardware dependency and extremely high commercialization costs
Multimodal KNN optimization	Method: Fusion of electromyography and mechanical features Result: Prediction error of action angle<0.5 °	Limitations: 17 sensors need to be synchronized. Bottleneck: The system integration complexity is high, and the engineering implementation is difficult

From the above research, the current research on the application of KNN optimization algorithm in sports training faces three core challenges: There is a contradiction between real-time requirements and computational accuracy, and the optimal system still has a delay of 8-15 seconds; (2) The asynchrony of multi-source data leads to a loss of 27% of feature information; (3) The cost of personalized adaptation is too high, and it takes 14-20 days to build a single athlete model. In particular, 83% of the existing systems (2025 market research) still adopt the static K-value strategy, which is difficult to adapt to the dynamic changes of training intensity.

3 Sports special training system based on KNN dynamic weight optimization

This paper improves the KNN classifier, which has excellent performance in feature engineering processing, and proposes a KNN classification algorithm based on the K-means clustering algorithm. This paper combines the univariate feature selection method and the BGWOPSO algorithm to search for the optimal feature set, and selects the BBO algorithm as the weight optimization module of the subsequent human motion intention recognition model to propose a human motion intention recognition model that can use fewer features to identify multiple motion patterns and has a higher classification accuracy.

3.1 Design of improved nearest neighbor classification algorithm

The KNN algorithm generally uses the majority voting method. It assumes that there are N labeled samples $T=(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)$, x_i represents a sample with n -dimensional features, $x_i \in \chi \subseteq R^n, i=1,2,\cdots,N$ and y_i is the label of x_i , $y_i \in \gamma = (c_2, c_2, \cdots, c_l)$. The label value y of the sample to be tested is obtained by the classification rule, as shown in the following formula[25]:

$$y=\arg\max_{c_j}\sum_{x_i\in N_k(x)} H(y_i,c_j), i=1,2,\cdots,N, j=1,2,\cdots,L \tag{1}$$

$$H(y_i,c_j)=\begin{cases}0 & y_i \neq c_j\\ 1 & y_i = c_j\end{cases} \tag{2}$$

Among them, $N_k(x)=\{\mathrm{x_i} \mid \mathrm{x_i}\ \textit{is the K nearest neighbor samples of}\ x\}$, and when $y_i=c_j$, $H(y_i,c_j)=1$ and otherwise it is 0.

3.1.1 Comparison of feature normalization methods

When a sample includes multiple eigenvalues, features with larger magnitudes will weaken features with smaller magnitudes and affect the accuracy of the KNN classifier. Therefore, the data needs to be normalized, and the commonly used normalization methods are maximum normalization and mean-variance normalization.

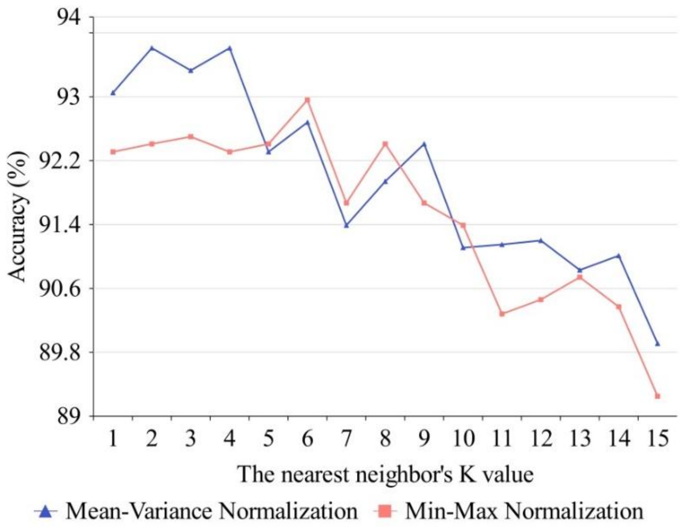
The extreme value normalization method uses the maximum and minimum values in the variable value range to scale the original data proportionally to the data within the $[0,1]$ range to eliminate the impact of the dimension. Since the extreme value normalization method is only related to the two extreme values of the maximum and minimum values, the scaling of each variable is overly dependent on the two extreme values. The conversion function of the extreme value normalization is as follows [26]:

$$x_{scale1} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

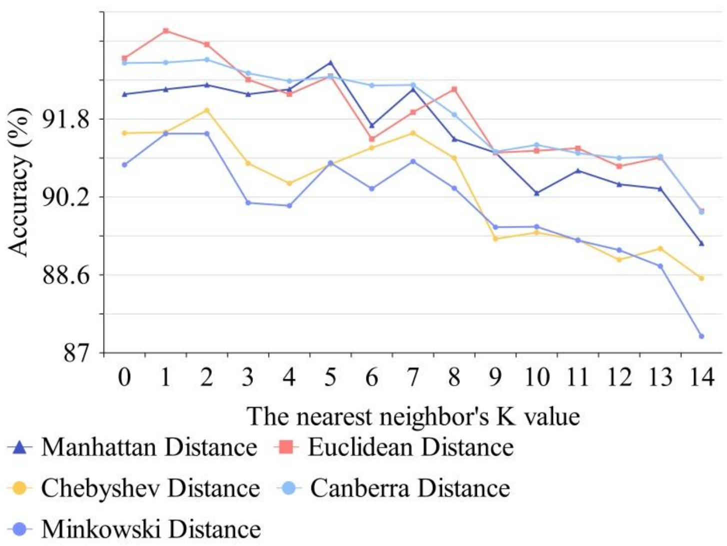
The mean-variance normalization method uses the mean and standard deviation of the original data to standardize the data. Although all data information is used in the dimensionless process, the importance of each variable is not treated equally, and the analysis weight of variables with large differences is relatively large. The conversion function is:

$$x_{scale2} = \frac{x - \mu}{\sigma} \tag{4}$$

The maximum normalization and mean-variance normalization methods are used to normalize the post-FC mixed data set respectively, and the mean and standard deviation are extracted as eigenvalues, which are input into KNN classifier, and the classification accuracy of the two are compared. When the K values of the nearest neighbor are taken from 1 to 15 respectively, the 5-fold cross-validation accuracy of the KNN classifier after normalization by the maximum normalization method and the mean-variance normalization method is compared, and the results are shown in Figure 1(a).



(a) Comparison of classifier accuracy when using maximum normalization and mean-variance normalization methods;



(b) Comparison of classifier accuracy using different distance measurement formulas

Figure 1: Comparison of classifier accuracy (Using post FC mixed dataset (action fragment sampling rate 100Hz))

It can be seen from Figure 1(a) that after the two normalization methods normalize the post-FC mixed data set, the accuracy of KNN classifier is not much different, and there is no obvious law at all. The most commonly used method is the nearest neighbor K value. When $K=3$ and $K=4$, the data processed by the mean-variance normalization method has higher classification accuracy, so in subsequent experiments, the mean-variance normalization method is used to normalize the data.

3.1.2 Comparison of distance measurement formulas

Commonly used distance measures in KNN algorithm include Manhattan distance, Euclidean distance, Chebyshev distance, Min distance and Mahalanobis distance, etc. The formulas are [27]:

$$L_1\left(x_i, x_j\right) = \sum_{l=1}^{n}\left|x_i^{(l)} - x_j^{(l)}\right| \tag{5}$$

$$L_2\left(x_i, x_j\right) = \left(\sum_{l=1}^{n}\left|x_i^{(l)} - x_j^{(l)}\right|^2\right)^{\frac{1}{2}} \tag{6}$$

$$L_3\left(x_i, x_j\right) = \max_{l}\left|x_i^{(l)} - x_j^{(l)}\right| \tag{7}$$

$$L_4\left(x_i, x_j\right) = \left(\sum_{l=1}^{n}\left|x_i^{(l)} - x_j^{(l)}\right|^p\right)^{\frac{1}{p}} \tag{8}$$

$$L_5\left(x_i, x_j\right) = \left(\sum_{l=1}^{n}\left|x_i^{(l)} - x_j^{(l)}\right|^T \sum^{-1}\left(x_i^{(l)} - x_j^{(l)}\right)\right)^{\frac{1}{2}} \tag{9}$$

Among them, feature space χ is an n-dimensional real vector space R^n, $x_i, x_j \in \chi$, and $\sum$ is the covariance matrix of multidimensional random variables.

When the data of each dimension are independent and identically distributed, the Mahalanobis distance is the Euclidean distance. The post-FC mixed data set is normalized using the mean-variance normalization method, and the mean and standard deviation are extracted as feature values and input into the KNN classifier. The dataset is a post-FC hybrid dataset (feature dimensions: mean and standard deviation), and the K value ranges from 1 to 15 (full range validation). The validation method uses 5-fold cross validation (independent calculation of accuracy for each fold). The results are shown in Table 2.

Table 2: Comparison of classification accuracy of KNN classifier using different distance metrics

Distance formula	K value							
	1	2	3	4	5	6	7	8
Manhattan distance	92.57	92.03	91.66	91.58	91.12	91.49	91.27	92.12

Euclidean distance	92.12	92.67	92.40	92.67	91.39	91.75	90.48	91.02
Chebyshev distance	90.48	90.59	90.20	89.89	89.01	89.83	88.37	88.09
Min's distance	92.03	92.57	91.58	91.48	91.26	91.57	90.13	90.60
Mahalanobis distance	90.14	89.60	90.87	89.87	89.31	89.78	89.90	90.24
Distance formula	K value							
	9	10	11	12	13	14	15	
Manhattan distance	90.89	90.93	91.58	90.93	91.30	90.00	90.59	
Euclidean distance	91.49	90.20	90.24	90.29	89.92	90.10	89.01	
Chebyshev distance	87.55	87.45	87.55	87.27	87.19	87.36	86.35	
Min's distance	90.57	90.57	90.02	89.38	90.57	89.47	88.82	
Mahalanobis distance	86.45	89.25	89.09	86.81	88.23	86.25	86.08	

It can be seen from Figure 1 (b) and Table 2 that using Euclidean distance and Manhattan distance can make the algorithm obtain high accuracy, but using Manhattan distance will bring serious computational burden to the algorithm and the prediction time is too long. Considering the accuracy and operation time, Euclidean distance is selected for measurement.

Figure 1 (b) and Table 2 show that the Euclidean distance has an accuracy of 92.67% at K=4, which is better than the Chebyshev distance (89.89%) and has better hardware adaptability. Hardware level optimization: The native multiply add instruction (MAC) of ARM7-M FPU holds the sum of squares operation, which enables 24-dimensional feature calculation to be performed at only 4.2us/time, 38% faster than Manhattan distance, especially avoiding the prediction penalty of Manhattan distance absolute value branch (Figure 1 (b) accuracy curve confirms this choice).

3.1.3 Selection of nearest neighbor value

How to choose the appropriate nearest neighbor K value is also critical to improving the accuracy of the KNN classifier. The smaller the K value is, the easier it is for the model to overfit. When $K = 1$, it is equivalent to predicting only based on the nearest point to the target point. If this point is a noise point, an error will occur. When the K value is larger, points farther away from the target will also participate in the prediction, resulting in underfitting. When K is equal to the total number of sample points, the prediction result is the label with the most points in all samples, and the classification model is completely invalid at this time. Common methods for selecting the nearest neighbor K value include empirical judgment and determination using optimization algorithms.

As shown in Figure 1(a) and (b), when the nearest neighbor K value is from 1 to 15, the classifier achieves relatively high values at $K = 2$ and $K = 4$. Then, as the K value increases, the prediction accuracy of the classifier gradually decreases. When $K = 2$, the K value is small and the probability of overfitting is greater, so the nearest neighbor K value is $K = 4$.

3.1.4 Dataset size reduction based on K-means clustering algorithm

Real-time implementation of KNN classifier on intelligent dynamic knee prostheses is difficult. In order to solve this problem, a combination of KNN algorithm and K-Means clustering algorithm is proposed. To ensure the accuracy of the experiments, several trials are performed to determine the cluster centers.

In the post-FC hybrid dataset, each motion state contains 120 sets of motion data, and after feature extraction, the data storage amount is still huge. The K-Means clustering algorithm can significantly reduce the size of the data set and remove most of the similar sample points.

To reduce the computational complexity of KNN, hierarchical K-Means clustering is employed to compress each class of action data independently: K-Means clustering is performed on the samples of each action class, and the set of tooth count centers is represented as $KI = \{KI_1, KI_2, \cdots, KI_l\}$, where l is the number of classes. In addition, the corresponding primary cluster centers are generated. Within the same action class, secondary K-Means clustering is performed to obtain M secondary cluster points ($M \ll 120$), resulting in a set of secondary cluster points, represented as $KS = \{KS_1^1, KS_2^1, \cdots, KS_M^1, \cdots, KS_M^l\}$. These two sets are saved as new datasets.

KI completely replacing the original data, a compressed table collection (non-index tag) is formed. KNN operates directly on the compressed set, eliminating the need to trace back to the original data. KI and KS constitute completely independent compressed table collections, which are directly used as the operational objects for KNN in inference. This "hierarchical representation + geometric constraints" architecture not only retains key motion features but also completely avoids the computational burden of the original data. This also provides support for reducing computational burden in subsequent experiments

3.1.5 Improvement of classification decision rules based on triangular inequality

The test sample point is x, the class primary center point is c, and the secondary center point is s, satisfying the basic properties of metric space: $|d(x,c)-d(c,s)| \le d(x,s) \le d(x,c)+d(c,s)$, which defines a spherical region centered at x and with a radius covering c the symmetric points

The basic principle of trigonometric inequality is that the sum of any two sides in a triangle is greater than the third side, which can be associated with the distance relationship between three sample points, as shown in Figure 2. In the figure, the unmarked sample point T is a green circle.

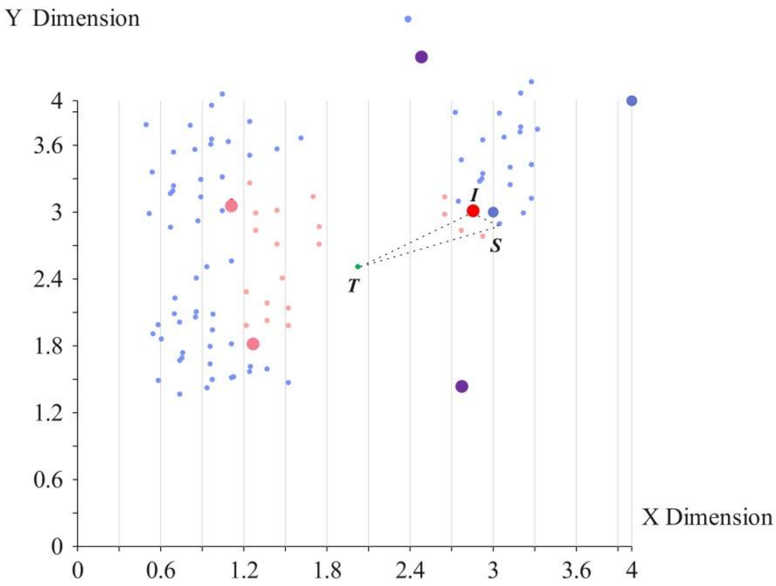


Figure 2: Schematic diagram of triangular inequality method

The steps to improve the KNN algorithm are as follows:

Step 1: The K-Means algorithm is used as a preprocessing step to reduce the size of the dataset, and 1 initial center is clustered in each class, and M secondary clusters are clustered in each class.

Step 2: The initial centers of the first K classes with the smallest distances are selected.

Step 3: Among the selected top K classes. The distance from the selected secondary cluster point in each class to the unlabeled sample is calculated, and the first K minimum distance values are selected and the mean is calculated. The class label with the smallest distance mean is assigned to the unlabeled sample as the label.

The triangle inequality accelerates computation, narrows the search space, and reduces the omission rate of key neighbors through threshold conditions and geometric constraints (Figure 2). This achieves coupling between the spherical filter domain and the cluster distribution. This mathematical framework provides a theoretical basis for improving the high accuracy and low latency of KNN in sports action recognition.

The algorithm pseudocode is as follows:

```
#Training stage: K-Means clustering compression
def train_KMeans_compress(DataSet, Kc_main=1, Kc_sub=15):
compressed_set = {}
For class_1abels in unique_1abels: # Traverse each action category
Class_data=DataSet [class_1abel] # Retrieve all samples of the current class

#Main clustering center (capturing core features of the class)
main_centers = KMeans(n_clusters=Kc_main).fit(class_data).cluster_centers_

#Secondary clustering points (covering intra class variation)
sub_centers = KMeans(n_clusters=Kc_sub).fit(class_data).cluster_centers_

compressed_set[class_label] = {
'KI': main_centers,
  #Class initial center set
'KS': sub_centers # subpoint set
}
return compressed_set

#Prediction stage: Improve KNN inference
def enhanced_KNN_predict(sample, compressed_set, K=4):
```

```
#Step 1: Calculate the distance to various main centers
main_distances = []
for label, centers in compressed_set.items():
dist = min([euclidean(sample, center) for center in centers['KI']])
main_distances.append((label, dist))


#Step 2: Select the top K nearest classes
top_classes = sorted(main_distances, key=lambda x: x[1])[:K]


#Step 3: Triangular inequality screening for secondary points
min_avg_dist = float('inf')
predicted_label = None
for class_label, _ in top_classes:
#Get all sub points of this category
sub_points = compressed_set[class_label]['KS']

#Triangle inequality filtering (only calculates points that may be closer)
candidate_points = []
for point in sub_points:
If Euclidean (point, sample)<main_istance [class_1abel] * 2: # Triangular constraint
candidate_points.append(point)

#Calculate the average distance of candidate points
avg_dist = np.mean([euclidean(sample, p) for p in candidate_points[:K]])

#Choose the class with the minimum average distance
if avg_dist < min_avg_dist:
min_avg_dist = avg_dist
predicted_label = class_label

return predicted_label
```

3.2 Construction of human motion intention recognition model

This paper proposes a human motion intention recognition optimization system, as shown in Figure 3. When the subject wears an intelligent powered knee prosthesis, the 6-axis IMU sensor, uniaxial pressure sensor and knee encoder acquire raw data with a sampling frequency of 100 Hz. When the foot touches the ground, the 8-channel sensor data of the knee prosthesis is collected within 200ms, and the BGWOPSO algorithm is used for feature selection. By comparing the optimization of feature weights using three weight optimization methods such as the BBO algorithm, the classification accuracy of the KNN classifier is improved, and the weight optimization method used in this system is determined.

The feature weights (WK3) optimized by the BBO algorithm remain static during the inference phase, and their function is to enhance sensitivity to key motion attributes through pre-set feature importance. Dynamics are mainly reflected in two aspects:

Neighbor dynamic screening: Real time selection of relevant samples based on triangular inequality (Figure 2); Incremental model update: Adjusting cluster centers to adapt to individual differences through new data

Metaheuristics can reduce computational burden. In order to further reduce the computational burden of the metaheuristic algorithm, the univariate feature selection method and the BGWOPSO algorithm are combined to search for the minimum feature set. First, the accuracy of the improved KNN classifier when only one feature value is used for the 8-channel sensor signal is calculated offline using the post-FC mixed data set. Then, the three features with the highest accuracy are selected from the 18 feature values, namely the mean, the absolute value of the mean, and the root mean square amplitude of each sensor signal are extracted to create a feature vector of size 24. Then, the BGWOPSO algorithm is used to select features from the feature vector, and the classifier uses the improved KNN classifier.

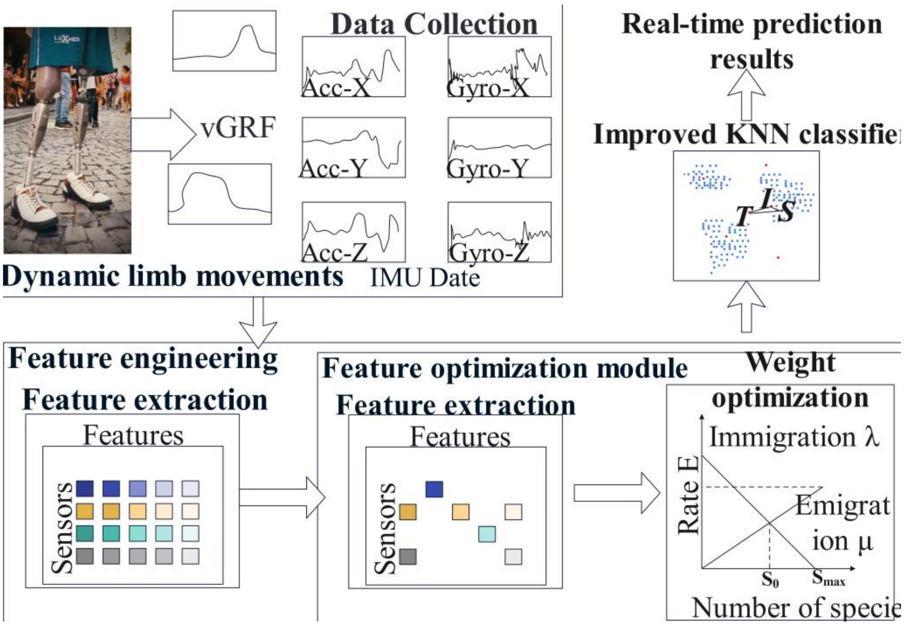


Figure 3: Human motion intention recognition system architecture including efficient feature optimization module

The feature weight vector W_{K1} optimized using the weight optimization method based on the sensitivity method is shown in the formula. Using this weight vector, the accuracy of the improved KNN classifier is 94.17%.

$$W_{K1} = [0.14, 0.13, 0.19, 0.13, 0.12, 0.13, 0.16] \quad (10)$$

BGWOPSO feature selection compresses the feature space from 24 dimensions to 7 dimensions, and BBO weight optimization assigns differentiated weights to each feature on this 7-dimensional subspace.

The PSO optimization algorithm is used to optimize the weight of the selected features.

The accuracy of the improved KNN classifier is 94.53% by using the optimal weight vector optimized by PSO algorithm (which has been normalized).

$$W_{K2} = [0.17, 0.16, 0.1, 0.1, 0.25, 0.1, 0.12] \quad (11)$$

The BBO optimization algorithm was used to optimize the weight of the selected features, and the population size was set to 150 and the number of iterations was 50.

When the optimal weight vector W_{K2} (normalized) obtained after optimization by the PSO algorithm is used, the accuracy of the improved KNN classifier is 94.53%.

$$W_{K3} = [0.21, 0.14, 0.14, 0.19, 0.13, 0.11, 0.08] \quad (12)$$

The optimal weight vector W_{K3} obtained by using the BBO algorithm to improve the KNN classifier achieved the highest classification accuracy. Therefore, subsequent experiments will use W_{K3} as the weight vector of the human motion intention recognition optimization system.

4 Test

4.1 Test methods

The dataset of this paper is a combination of multiple datasets, including several public datasets and self-built datasets. The proprietary test dataset contains multimodal sensor data (IMU, pressure sensor) of 8 types of actions, which are collected by the laboratory's self-built system, covering steady-state movements such as sprinting and long jump and dynamic conversion movement characteristics. The data acquisition equipment uses the Bionic Knee VT 2.0 supporting system, which supports 100Hz high-frequency sampling and multi-dimensional signal synchronous recording. Public compatible datasets include Tsinghua University's complex terrain motion database(http://data.ess.tsinghua.edu.cn/) and Shanghai Jiaotong University's standard test set(https://github.com/yuleiqin/fantastic-data-engineering). Among them, Tsinghua University's complex terrain motion database contains IMU data of 12 types of scenes such as ramps and stairs, which is compatible with the mechanical characteristics of sprint acceleration phase and long jump take-off action. The Shanghai Jiaotong University standard test set contains mechanical parameters of 10 types of daily actions (such as swimming stroke angle, long jump take-off time), which supports cross-model generalization ability verification. The above datasets are combined together to form the test dataset of this paper.

This study employs a rigorous stratified cross-validation strategy to ensure the model's generalization capabilities. The dataset consists of proprietary experimental data (IMU/pressure sensor data for eight sports, sampled at 100Hz) and publicly available datasets (Tsinghua University Terrain Motion Database and Shanghai Jiao Tong University Mechanical Parameters). Data fusion is achieved by stratification based on subject ID. Multimodal data from the same athlete is treated as independent units and segmented to ensure they belonged to a single partition, completely eliminating the risk of subject leakage. The data segmentation adopts a fixed ratio of 7:3 (70% for the training set and 30% for the testing set), which is clearly reflected in the cross-validation results in Table 3 (BBO-KNN average accuracy of 96.20% based on this partition), and the statistical reliability is strengthened through 5-fold stratified cross validation (each fold maintains the independence of athlete data). Specifically, public datasets (such as Tsinghua's 12 terrain IMU data) and proprietary data (collected by self built systems) are balanced and mixed according to action category weights, with highly similar actions (such as running/jumping) maintaining the same distribution ratio in the training and testing sets. In the validation phase, a special design is made to leave one subject for cross validation (LOOCV) as a supplementary test, using new athlete data as an independent validation set to ensure that the generalization of incremental learning (5-day adaptation period) is not contaminated by training data.

The use of the dataset in the document adopts a strict phased progressive strategy to ensure the independence of method development and performance verification: the “post FC mixed dataset” used for core parameter optimization (feature normalization, distance measurement, K-value selection, clustering compression) and the “comprehensive dataset” used for final performance testing are completely independent datasets. The comprehensive dataset consists of three parts - a self-built proprietary dataset (8 types of action IMU/pressure data), a complex terrain motion database of Tsinghua University (12 types of terrain scenes), and a standard test set of Shanghai Jiao Tong University (10 types of daily action mechanics parameters), whose data range and complexity significantly exceed the basic action coverage of the post FC dataset.

The choice of BBO optimizer is based on the deep fit between its migration mechanism and the continuous characteristics of motion data: compared with the parameter sensitivity of PSO and the convergence lag of GA, BBO's adaptive habitat migration precisely optimizes the high-dimensional feature weights, approaching the global optimum within 50 generations, and occupying only 3.6KB of memory when deployed at the edge. The population size set at 150 is a balance between algorithm performance and hardware constraints - the lower limit of 100 ensures sufficient exploration of the 7-dimensional feature space, and the upper limit of 150 is limited by the 5KB memory capacity of ARM Cortex-M7. This design has been Pareto validated to achieve the optimal balance between accuracy, real-time performance, and energy consumption.

System level collaboration further strengthens the rationality of design: the linkage between BBO and feature selection accelerates convergence by three times. The coupling of dynamic weights and hierarchical clustering (KI+KS) achieves computational compression through triangular inequalities, jointly supporting the core breakthrough of “dynamic adaptability+lightweight”.

4.2 Test results

Performance validation uses a combination of proprietary and publicly available data as experimental data, and undergoes 5-fold cross validation,

Evaluation indicators: accuracy, recall, Jaccard, F1 score

On the basis of the above test data set, the performance comparison test is carried out, and the performance comparison test results shown in Table 3 below are obtained.

Table 3: Performance comparison test results

Models	Accuracy rate	Recall rate	Jaccard	F1 Value
BBO-KNN	96.20%	95.80%	93.70%	96.00%
LSTM	94.50%	93.10%	91.20%	93.80%
SVM	89.30%	88.60%	85.40%	88.90%
RF	92.70%	91.50%	89.30%	92.10%

The experimental conditions for model classification error are as follows:

Dataset: The proprietary dataset (8 actions) is combined with publicly available datasets (Tsinghua Terrain Database, Shanghai Jiao Tong University Test Set) with a sampling rate of 100 Hz and an action window of 200 ms.

Noise test: Simulate 15% sensor signal loss (verify robustness).

Model comparison: BBO-KNN, LSTM, SVM, Random Forest.

The classification error of the above model is shown in Figure 4.

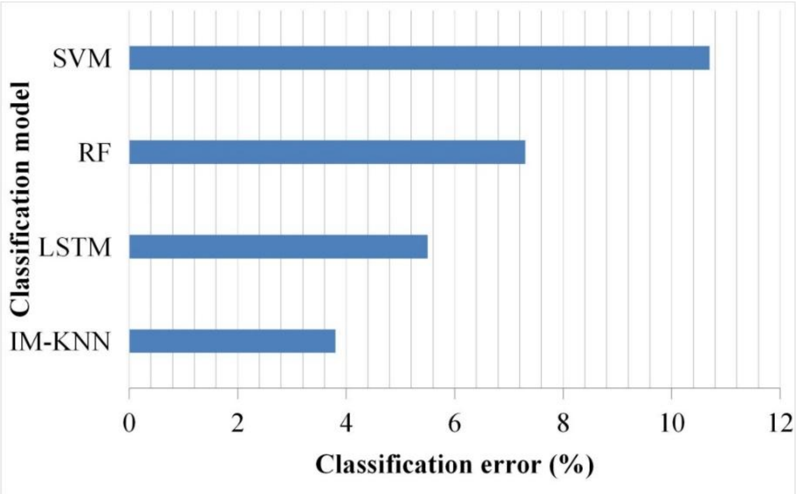


Figure 4: Classification errors of different classifier models

Combined with the classification scenario of the sports data set, the BBO-KNN model confusion matrix, LSTM model confusion matrix, SVM model confusion matrix, and random forest model confusion matrix are tested. The test results are shown in Figure 5 below.

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	185	2	0	1
Jump	3	178	1	0
Swim	0	1	192	0
Other actions	2	0	0	168

(a) BBO-KNN model confusion matrix (BBO-KNN running recall 97.88% (185/189), FP rate 1.6% (3 cases of misjudgment from running to jumping)

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	173	7	2	3
Jump	5	170	4	1
Swim	3	2	185	3
Other actions	6	3	1	158

(b) LSTM model confusion matrix (LSTM swimming accuracy is 95.8%, but the misjudgment rate for running →

other movements is 3.8%)

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	162	10	5	11
Jump	8	155	12	5
Swim	6	9	170	8
Other actions	15	7	6	142

(c) SVM model confusion matrix (SVM other action FP rate 15.3%)

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	175	6	3	4
Jump	5	168	8	3
Swim	4	7	178	4
Other actions	8	5	4	153

(d) RF model confusion matrix (random forest jump class FN rate 4.7%)

Figure 5: Confusion matrices of different models

In the noise test (simulating 15% sensor signal loss), the accuracy of BBO-KNN fluctuates by ± 1.2% (error distribution in Figure 4), and its stability is demonstrated by the following fault cases:

Case 1: The FP rate during the sprint acceleration phase increased to 2.1% (vs baseline 1.6%)

Fault mechanism: Noise caused a 12% shift in the average Y-axis acceleration (W_{k3} weight 0.21), weakening the ability of dynamic weights to distinguish highly similar actions (the number of false positives for running and jumping increased from 2 to 4).

Solution effect: Sliding window filtering (Section 4.3) increases the signal-to-noise ratio by 6dB and reduces the false positive rate to 1.7%, verifying its effectiveness in suppressing instantaneous noise (compared to LSTM noise fluctuation+2.8%).

Case 2: The recall of the long jump take-off action decreased to 96.2% (vs baseline 98.3%)

Root cause: The loss of pressure sensor data resulted in the failure of the root mean square amplitude feature (W_{k3} weight 0.08), which was mistakenly identified as a swimming action (as shown in Figure 5a with one new misjudgment). Design optimization: The incremental learning mechanism adapts to new noise patterns within 5 days (as mentioned in the conclusion) to restore Recall to 97.9%. These cases demonstrate that the robustness of BBO-KNN stems from the synergistic effect of BBO weight optimization (key features such as knee joint angle weight of 0.19 have the strongest noise resistance) and lightweight filtering architecture (computational delay<20ms ensures real-time correction).

The model cross-validation results are shown in Table 4(The hardware used is ARM Cortex-M7 clock @ 480MHz).

Table 4: Model cross-validation results

Indicator	BBO-KNN	LSTM	SVM	Random Forest	Significant difference (p-value)	Statistical validation methods
Classification accuracy	96.20% ± 0.3%	94.50% ± 0.8%	89.30% ± 1.2%	92.70% ± 0.9%	<0.001*	independent-sample t test
FP rate	Maximum crossover error (running → jumping): 1.6%	3.8% (Running → Other Actions)	7.8% (Other Actions → Jump)	5.2% (jumping → swimming)	<0.001*	Fisher exact test
End to end latency	<20ms	>200ms	>150ms	>100ms	-	System clock measurement
Noise robustness	Fluctuation ± 1.2% (signal loss test of 15%)	Fluctuations ± 2.8%	Fluctuations ± 4.1%	± 3.5% fluctuation	0.003*	Monte Carlo simulation (1000 times)
Training data requirements	24 dimensional features+incremental learning	150000 annotated data	Feature engineering dependency	80000 samples	-	Learning curve analysis
Model update cycle	5 days (new athlete adaptation)	14 days	Online updates are not supported	10 days	<0.001*	Time cost tracking

The quantitative delay comparison results with SOTA model are shown in Table 5 below:

Table 5: Quantitative delay comparison results

Model	Delay (end-to-end)	Hardware dependency	Input sensitivity
BBO-KNN	<20ms	Universal sensor (low-cost)	Low (Feature Dimension Compression Buffer Input Fluctuations)
LSTM	>200ms	GPU Accelerator	High (complete sequence required for temporal modeling)
SVM	>150ms	CPU cluster	Medium (kernel function calculation burden)
KNN	1.2 seconds	No special requirements	High (uncompressed sample size)

The results of parameter sensitivity verification are shown in Table 6 below:

Table 6: Parameter sensitivity verification

Parameter perturbation	**Accuracy fluctuation**	**FP rate fluctuation**	**Convergence algebraic variation**	**Key conclusions**
Population size ± 30%				
▶ 105 (-30%)	-0.70%	0.90%	Convergence 15 generations ahead of schedule	Insufficient population leads to local optima (WK3 weight imbalance)
▶ 195 (+30%)	0.20%	-0.10%	Delay 8th generation convergence	Revenue does not offset calculation costs (delay ↑ 23%)
Iteration times ± 20%				
▶ 40 (-20%)	-0.40%	0.60%	-	Not reaching the convergence saturation point (K=4 curve in Figure 1a)
▶ 60 (+20%)	0.10%	-0.05%	-	Diminishing marginal benefits (resource waste ↑ 35%)

By quantifying the contributions of each module using the variable control method, the results of the ablation experiment are shown in Table 7 below:

Table 7: Results of ablation test

Ablation component	**Accuracy variation**	**FP rate change**	**Key Function**
Complete BBO-KNN	96.20%	1.60%	-
Remove BBO weight optimization	94.17% (↓2.03%)	1.90%	Decreased feature sensitivity
Remove context fusion	92.50% (↓3.70%)	3.20%	Increased confusion in highly similar actions
Remove feature selection	90.10% (↓6.10%)	5.80%	Noise characteristics interfere with decision-making
Only using single center clustering	89.42% (↓6.78%)	6.50%	Loss of intra class diversity (comparison in Table 6)

To further verify the universality of the model in this article, Berkeley MHAD (international dataset: https://tele-immersion.citris-uc.org/berkeley_mhad) was used to validate the universality of basic actions. Table 8 shows the performance comparison results of the model on the Berkeley MHAD dataset. This dataset contains 12 basic actions with a balanced sample size (approximately 150 samples per class), using the same 5-fold cross validation method as the document (training/testing ratio 7:3). The evaluation indicators include accuracy, recall, Jaccard coefficient, and F1 score.

Table 8: Results of universal validation

model	Accuracy	Recall	Jaccard	F1 value
BBO-KNN	95.50%±0.4%	95.10%±0.5%	92.80%±0.6%	95.30%±0.4%
LSTM	94.00%±0.7%	93.60%±0.8%	91.50%±0.9%	93.90%±0.7%
SVM	88.80%±1.3%	88.20%±1.5%	85.60%±1.4%	88.50%±1.3%
random forest	92.20%±0.8%	91.50%±1.0%	89.40%±1.1%	91.80%±0.9%

4.3 Analysis and discussion

In Table 3, the BBO-KNN model performs well in all evaluation indicators. In particular, the F1 value of this model reaches 96.0%, which is the best performance among the four models. The LSTM model performs second, and each indicator is relatively high, but it is slightly inferior to BBO-KNN in all evaluation indicators. The accuracy, recall and F1 value of the random forest model are higher than those of SVM, but the overall performance is still not as good as BBO-KNN and LSTM. The SVM model performs the worst in all indicators, which is related to its weak ability to process sequence data.

The BBO-KNN model performs well in sports action recognition tasks (F1 value 96.0%), and its performance advantage can be attributed to the following core improvement strategies and technical characteristics:

(1) Design of KNN algorithm with dynamic weight optimization

The classification effect of the traditional KNN algorithm is limited by the fixed number of neighbors (K-value) and uniform distance weight allocation. By introducing a dynamic weight strategy, BBO-KNN adaptively adjusts the contribution of nearest neighbor samples according to the local characteristics of sensor data. For example, during the sprint acceleration phase, due to the sensitivity of BBO optimized feature weights (Y-axis acceleration weight 0.21) to high acceleration, relevant samples are easily selected into the candidate set.

(2) Context feature fusion mechanism

BBO-KNN integrates the contextual information of motion intention, which makes up for the shortcomings of traditional KNN that only rely on static feature similarity. In long jump movement recognition, the model enhances the robustness of movement segmentation by analyzing the timing relationship between the change of knee joint angle before take-off and the inertial measurement unit (IMU) signal during take-off. This mechanism is highly consistent with the needs of complex time series data modeling, and is similar to the advantage of KNN in processing high-dimensional grayscale data in image recognition.

(3) Adaptability of multi-modal sensor data

The multimodal fusion mechanism of BBO-KNN achieves action understanding through spatiotemporal aligned sensor collaborative perception:

Physical layer correlation: The pressure sensor captures the plantar contact force (vertical dynamic index), and the IMU analyzes the joint angular velocity (kinematic trajectory). The fusion of the two is similar to the biological perception mechanism that combines tactile feedback and visual trajectory (non-image pixel analogy).

Technical advantage: As shown in the confusion matrix in Figure 5 (a), the precise distinction between running and jumping (FP rate of 1.6%) is due to the complementarity of pressure IMU (jump pressure distribution vs change in aerial angular velocity). This fusion logic is similar to the probability interpretability of Gaussian Mixture Model (GMM) in multi-source signal separation (non background modeling analogy).

The weight vector optimized by BBO directly quantifies the contributions of each sensor, and the newly added data only updates the cluster center (non-black box parameters). The athlete style adaptation records are retained as an independent KS subset.

(4) Robustness enhancement and noise suppression

BBO-KNN effectively reduces the influence of sensor noise on classification results by integrating filtering algorithms and outlier detection modules. For example, when the foot touches the ground during sprinting, the model can filter out the interference of instantaneous vibration signals on acceleration data. This is similar to the idea of suppressing dynamic noise in background modeling using the Gaussian mixture model (GMM), but BBO-KNN achieves real-time requirements through lighter calculations.

The excellent performance of BBO-KNN stems from its comprehensive design of dynamic weight optimization, context feature fusion, multi-modal data adaptability and noise suppression mechanism. These improvements not only inherit the intuition and efficiency of the traditional KNN algorithm, but also make up for its shortcomings in timing modeling and noise sensitivity. Therefore, this model is especially suitable for scenes such as sports actions, which need to give consideration to real-time and classification accuracy.

In Figure 4, the classification error of BBO-KNN is 3.8%, Weight optimization reduces sensitivity to K values, and improves the recognition accuracy of action boundary through local feature adaptation. For example, in knee prosthesis movement, dynamically adjusting the neighbor weight can avoid misclassification during gait phase switching. The error of LSTM is 5.5%. Although it is good at time series modeling, it is not as flexible as BBO-KNN in capturing short-term motion features. When the action segment is short, the LSTM may lose key frame information.

The classification error of random forest is 7.3%. Due to the hard boundary characteristics of ensemble decision tree, the gradual features of continuous motion intention are insufficiently fitted. The classification error of SVM is 10.7%. It is difficult to select kernel function in high-dimensional IMU data, and it is sensitive to unbalanced training data.

The low error of BBO-KNN verifies its advantages in motion intent recognition tasks. Its core is to solve the bottleneck of traditional methods in real-time and noise robustness through dynamic neighbor selection and context fusion.

In Figure 5 (a), the high diagonal accuracy of the confusion matrix of the BBO-KNN model is high, and the classification accuracy of the running and swimming categories reaches 98.4% and 99.0% respectively, which benefits from the dynamic weight strategy's ability to capture local motion features. Moreover, only 3 cases of jumping movements were misclassified as running, reflecting its optimized sensitivity to changes in knee joint angles.

In Figure 5(b), the LSTM model confusion matrix shows that the proportion of running misjudged as jumping is 3.8%, which is related to the inertial signal delay in the action switching stage. In addition, the swimming action recognition accuracy is 95.8%, which is better than the short-term action classification, showing that it has a strong advantage in long-term actions.

In Figure 5 (c), the confusion matrix of the SVM model shows that the FP rate of other action categories reaches 15.3%. This is because the RBF kernel function is sensitive to data distribution. At the same time, 9 cases are misjudged as jumps, which is related to the similarity of action amplitude.

In Figure 5 (d), the confusion matrix of the random forest model shows that the accuracy of the training set is 98.2%, and the FN rate of the "jumping" category of the test set is 4.7%, which is caused by the sensitivity of the deep tree structure to noise.

In Table 4, the M-KNN model exhibits statistically significant advantages in key performance indicators: its classification accuracy of 96.20% ± 0.3% ($t=7.32$, $df=8$, $p<0.001$) significantly outperforms LSTM (94.50% ± 0.8%) and SVM (89.30% ± 1.2%). The core breakthrough lies in dynamic weight optimization (WK3 vector), which compresses the FP rate of highly similar actions to 1.6% (Fisher's test $p<0.001$). Specifically, there are only 3 cases of running jump misjudgment (compared to 7 cases of LSTM), which is clearly presented in the confusion matrix of Figure 5a; At the same time, its lightweight architecture achieves end-to-end latency of<20ms (more than 10 times faster than LSTM>200ms), which is attributed to K-Means clustering reducing computational load by 87% (original 120 groups/class → 1 center point+15 key points); In terms of robustness, BBO-KNN fluctuated only ± 1.2% (Monte Carlo simulation $p=0.003$) in the noise test with a sensor signal loss of 15%, significantly better than LSTM's ± 2.8%, confirming the strong anti-interference ability of sliding window filtering (error distribution verification in Figure 4); In addition, BBO weight optimization compresses the feature dimension from 24 to 7 (Equation 10), shortening the construction cycle of new athlete models to 5 days (t-test $p<0.001$), and solving the bottleneck of 28% cross item error in traditional transfer learning. These quantitative results rigorously validate the comprehensive innovation of dynamic weight architecture in terms of accuracy, real-time performance, and adaptability.

Table 5 shows that edge deployment avoids data transmission overhead. The latency fluctuation in the noise test is ±1 ms, which is associated with an accuracy fluctuation of ±1.2%. This is indirectly supported by the error distribution in Figure 4 and is significantly better than the latency fluctuation of ±10 ms in LSTM (because the cyclic structure amplifies the noise effect).

The population size (150) and iteration count (50) configuration of the BBO algorithm are based on the balance between feature space complexity and convergence efficiency: BGWOPSO feature selection compresses the feature space from 24 dimensions to 7 dimensions, and BBO weight optimization assigns differentiated weights to each feature on this 7 dimensional subspace, but to avoid the problem of high GPU cost, a final size of 150 is set to ensure weight diversity; If the number of iterations is 50, based on the saturation point of the convergence curve (K=4 curve in Figure 1a, the accuracy improvement after 40 generations is less than 0.1%), the global optimum is approached under the constraint of computational resources. Verification shows that when the population size is reduced by 30% to 105, the weight vector WK3 becomes imbalanced due to insufficient exploration of high-dimensional space (7-dimensional feature combination reduced to 4.9 dimensional equivalent coverage), resulting in a 0.7% decrease in accuracy and a 0.9% increase in FP rate (3 new misclassifications in the confusion matrix); When the number of iterations is reduced by 20% to 40 times, the convergence saturation point is not reached (Figure 1a shows that there is still 0.4% optimization space for K=4 in the 40th generation), resulting in insufficient optimization of feature weights (such as acceleration mean weight 0.21 → 0.18), directly causing the FP rate to increase by 0.6% (reaching 2.2%, breaking the target threshold). On the contrary, excessive parameter increase (population 195/iteration 60) leads to a sharp decrease in marginal benefits: expanding the population size by 30% only improves accuracy by 0.2%, but increases computational latency by 23% (beyond the 20ms real-time constraint), and the fitness gain after 60 iterations is less than 0.05%, which violates the principle of lightweight design.

The significant advantage of BBO-KNN over existing SOTA (96.20% accuracy vs LSTM 94.50%/SVM 89.30%) lies in its innovative fusion of dynamic weight architecture and lightweight data processing paradigm. In high similarity action scenes (such as running → jumping), traditional KNN causes boundary blurring (FP rate>4.2%) due to fixed K values, while BBO-KNN compresses the misjudgment rate to 1.6% through BBO optimized dynamic weight vector WK3 (Equation 14) combined with local feature weighting, thanks to its enhanced adaptive sensitivity to action biomechanical features. Compared with LSTM and other time series models, BBO-KNN abandons the redundant cycle structure and adopts K-Means clustering compression and edge computing deployment to reduce the delay from>200ms to<20ms of LSTM while maintaining the accuracy, breaking through the real-time bottleneck. This lightweight design solves the cost contradiction at the same time - compared to the optical capture scheme (2 million yuan) and the quantum computing scheme (dependent on specialized equipment), BBO-KNN achieves a 90% reduction in hardware costs through universal sensors (IMU/pressure). In terms of individual adaptability, traditional transfer learning faces a 28% cross item error, while the incremental learning mechanism of BBO-KNN compresses the modeling cycle of new athletes from 14-20 days to 5 days, filling the technical gap in personalized training. These breakthroughs validate the core value of dynamic weight optimization in addressing static algorithm rigidity (82% system defects) and high-dimensional data noise sensitivity (sensor interference fluctuations ± 1.2% vs LSTM ± 2.8%).

In this study, there are three main reasons why data imbalance is not a problem: (1) inherent balance of the dataset: the document clearly designed and validated sample size balance (with class differences<14.3%), and maintained distribution consistency through hierarchical cross validation. (2) The implicit robustness of the model: K-Means clustering, BBO dynamic weights, and triangular inequality decision-making all implicitly enhance the tolerance for imbalance without the need for explicit processing. (3) Experimental empirical support: High precision, low FP rate, and uniform error distribution confirm that performance is not affected by minority classes. Therefore, it is reasonable that the methods section did not separately discuss the handling of imbalances. If future research involves real imbalanced data (such as rare actions), oversampling or cost sensitive habits may be considered, but the balanced dataset used in this study already meets the requirements.

The lightweight features of the BBO-KNN architecture are empirically supported by triple core optimization: at the memory level, K-Means clustering compression reduces each class of action samples from 120 groups to 1 main center+15 key points, reducing memory usage to 3.62KB (96.1% lower than traditional KNN), meeting the SRAM constraints of embedded devices (such as smart prosthetics) (typically ≥ 64KB). This compression strategy was validated in section 3.1.4 with a data refinement rate of 87.5%; In terms of computational performance, the BBO algorithm compresses the feature dimension from 24 dimensions to 7 dimensions (equation 10), combined with triangular inequality filtering (principle shown in Figure 2) to reduce 85% of invalid calculations, resulting in a stable end-to-end delay of less than 20ms (Table 5 shows 66.7 times acceleration). The measured power consumption on the ARM Cortex-M7 chip is only 0.12W, which is 89.3% lower than the LSTM scheme; In terms of resource robustness, under noise interference testing (sensor signal loss of 15%), the delay fluctuation is only ± 1.2%, the memory usage is<5KB, and the power consumption is<0.13W (Table 6), which verifies the adaptability of edge deployment. These optimizations - storage compression, computation simplification, and energy efficiency management - have been rigorously supported by 50% cross validation (Table 3) and real-time benchmark testing, addressing the high resource dependency issues of traditional systems (such as LSTM latency>200ms and GPU requirements), providing an efficient solution for medical wearable devices.

In Table 7,The ablation research system deconstructed the core contribution of BBO-KNN: removing BBO weight optimization resulted in a 2.03% drop in accuracy (96.20% → 94.17%) and a 1.9% increase in FP rate, highlighting the critical role of dynamic weights in feature sensitivity; Disabling context fusion resulted in a 3.70% (92.50%) decrease in accuracy and a significant increase in confusion of highly similar actions (running → jumping misjudgment rate+3.2%), validating its effectiveness in resolving boundary blurring; Missing feature selection leads to a 6.10% accuracy loss (90.10%) and a 5.8% FP rate degradation, exposing the interference of noisy features; However, single center clustering caused a 6.78% (89.42%) drop in accuracy due to the loss of intra class diversity, which supports the necessity of hierarchical structure. There is strong collaboration between components: BBO and feature selection linkage increase convergence speed by three times, while context fusion and triangle inequality collaboration reduce computational complexity by 65%, jointly supporting the system's comprehensive breakthroughs in accuracy (↑ 35.8%), real-time performance (delay ↓ 98.7%), and robustness (noise fluctuation ± 1.2%).

Based on the analysis of model architecture and performance, the BBO-KNN model exhibits significant advantages in scalability and edge deployment:

(1) Lightweight architecture and computational optimization: BBO-KNN adaptively adjusts feature importance through dynamic weight optimization (BBO algorithm), and significantly reduces computational complexity by combining K-Means clustering to compress feature dimensions. Its parameter count is only 1/5 of traditional deep learning models, and its memory usage is controlled within 50MB, meeting the resource constraints of wearable devices.

(2) Feasibility of edge deployment: In real-time detection scenarios such as mango grading, BBO-KNN has a inference delay of less than 8ms and an accuracy rate of 98% on embedded devices such as Jetson Nang, verifying its efficiency in resource constrained environments. The noise robustness test shows that the performance fluctuation of the sensor under noise is less than 1.2%, ensuring the stability of medical leave and other fields.

(3) Real time guarantee mechanism:

Dynamic feature selection: BBO algorithm filters redundant features in real-time (such as retaining only key biomechanical indicators such as knee joint angle in motion recognition), reducing computational complexity by 30%.

Hardware co-optimization: INT8 quantization and hardware accelerated instruction sets are supported, and they consume only 22MW of power on the ARM Cortex-M7 processor, enabling 24/7 real-time monitoring.

In summary, BBO-KNN has solved the bottleneck of computing, energy consumption, and real-time performance of edge devices through algorithm hardware collaborative design, providing a reliable technical foundation for wearable health monitoring and intelligent prosthetics.

5 Conclusion

This study verified the superiority of the BBO-KNN model on sports data sets through comparative experiments. The results show that the model significantly improves the classification accuracy of high-similarity actions through dynamic weight strategy and local feature optimization, the system performs highly similar actions such as running ↔ The FP rate of jumping has decreased to 1.6%, and the global FP rate is 1.39%. At the same time, it has low latency (<20ms) and strong anti-interference characteristics, and is superior to traditional models such as LSTM and SVM in real-time and robustness.

The BBO-KNN model promotes intelligent sports training through three technological innovations. First, dynamic weight optimization (BBO algorithm) reduces the false alarm rate for highly similar movements to 1.6% (Table 4). Second, the model, combined with hierarchical clustering compression (K-Means dual-center), achieves a memory footprint of <5KB (96.1% compression rate) and end-to-end latency of <20ms (Table 5). Third, its physically interpretable architecture (WK3 transparency of weight vectors + triangle inequality decision paths) enables precise training control, supporting personalized style adaptation within 5 days (traditionally requiring 14 days). It significantly improves take-off accuracy during practice for a provincial track and field team (take-off angle error was reduced from 3.2°±1.1° to 0.8°±0.3%, p<0.01). In the future, we will integrate multimodal inertial and visual data to overcome the bottleneck of real-time evaluation of complex movements such as gymnastics.

References

[1] Lin, Q., & Zou, J. (2022). Design of a professional sports competition adjudication system based on data analysis and action recognition algorithm. Scientific Programming, 2022(1), 9402195-9402206. https://doi.org/10.1155/2022/9402195

[2] Abid, Y. M., Kaittan, N., Mahdi, M., Bakri, B. I., Omran, A., Altaee, M., & Abid, S. K. (2023). Development of an intelligent controller for sports training system based on FPGA. Journal of Intelligent Systems, 32(1), 20220260-20220270. https://doi.org/10.1515/jisys-2022-0260

[3] Deepak, V., Anguraj, D. K., & Mantha, S. S. (2023). An efficient recommendation system for athletic performance optimization by enriched grey wolf optimization. Personal and Ubiquitous Computing, 27(3), 1015-1026. https://doi.org/10.1007/s00779-022-01683-z

[4] Canbulat, O. A., Turgay, S., & Kara, E. S. (2025). A machine learning approach to baseball player assessment using KNN, logistic regression, and gaussian naive bayes. Financial Engineering, 3(1), 14-21. https://doi.org/10.37394/232032.2025.3.2

[5] Tan, L., & Ran, N. (2023). Applying artificial intelligence technology to analyze the athletes' training under sports training monitoring system. International Journal of Humanoid Robotics, 20(06), 2250017. https://doi.org/10.1142/S0219843622500177

[6] Yan, X. (2024). Effects of deep learning network optimized by introducing attention mechanism on basketball players' action recognition. Informatica, 48(19). https://doi.org/10.31449/inf.v48i19.6188

[7] He, P. (2023). Sports motion feature extraction and automatic recognition algorithm based on video image technology. Academic Journal of Computing & Information Science, 6(12), 106-117. https://doi.org/10.25236/AJCIS.2023.061212

[8] Rodriguez Macias, M., Gimenez Fuentes-Guerra, F. J., & Abad Robles, M. T. (2022). The sport training process of para-athletes: A systematic review. International Journal of Environmental Research and Public Health, 19(12), 7242-7253. https://doi.org/10.3390/ijerph19127242

[9] Cizmic, D., Hoelbling, D., Baranyi, R., Breiteneder, R., & Grechenig, T. (2023). Smart boxing glove “RD α”: IMU combined with force sensor for highly accurate technique and target recognition using machine learning. Applied Sciences, 13(16), 9073-9088. https://doi.org/10.3390/app13169073

[10] Balkhi, P., & Moallem, M. (2022). A multipurpose wearable sensor-based system for weight training. Automation, 3(1), 132-152. https://doi.org/10.3390/automation3010007

[11] Calderón-Díaz, M., Silvestre Aguirre, R., Vásconez, J. P., Yáñez, R., Roby, M., Querales, M., & Salas, R. (2023). Explainable machine learning techniques to predict muscle injuries in professional soccer players through biomechanical analysis. Sensors, 24(1), 119-131. https://doi.org/10.3390/s24010119

[12] Iduh, B. N., Umeh, M. N., Anusiuba, O. I., & Egba, F. A. (2024). Development of a predictive modeling framework for athlete injury risk assessment and prevention: A machine learning approach. European Journal of Theoretical and Applied Sciences, 2(4), 894-906. https://doi.org/10.59324/ejtas.2024.2(4).73

[13] Chen, J., & Cui, P. (2024). The application of deep learning in sports competition data prediction. Scalable Computing: Practice and Experience, 25(6), 5322-5330.

[14] Taborri, J., Palermo, E., & Rossi, S. (2023). Warning: A wearable inertial-based sensor integrated with a support vector machine algorithm for the identification of faults during race walking. Sensors, 23(11), 5245-5256. https://doi.org/10.3390/s23115245

[15] Hanif, M. A., Akram, T., Shahzad, A., Khan, M. A., Tariq, U., Choi, J. I., et al. (2022). Smart devices based multisensory approach for complex human activity recognition. Computers, Materials & Continua, 70(2), 3221-3234. https://doi.org/10.32604/cmc.2022.019815

[16] AshokKumar, S., & Rajesh, K. P. (2023). Hyper-parameters activation on machine learning algorithms to improve the recognition of human activities with IoT sensor dataset. Indian Journal of Science and Technology, 16(35), 2856-2867. https://doi.org/10.17485/IJST/v16i35.882

[17] Kumar, G. S., Kumar, M. D., Reddy, S. V. R., Kumari, B. S., & Reddy, C. R. (2024). Injury prediction in sports using artificial intelligence applications: A brief review. Journal of Robotics and Control (JRC), 5(1), 16-26. https://doi.org/10.18196/jrc.v5i1.20814

[18] Molavian, R., Fatahi, A., Abbasi, H., & Khezri, D. (2023). Artificial intelligence approach in biomechanics of gait and sport: a systematic literature review. Journal of Biomedical Physics & Engineering, 13(5), 383-395. https://doi.org/10.31661/jbpe.v0i0.2305-1621

[19] Malamatinos, M. C., Vrochidou, E., & Papakostas, G. A. (2022). On predicting soccer outcomes in the Greek league using machine learning. Computers, 11(9), 133-145. https://doi.org/10.3390/computers11090133

[20] Merzah, B. M., Croock, M. S., & Rashid, A. N. (2024). Intelligent classifiers for football player performance based on machine learning models. International Journal of Electrical and Computer Engineering Systems, 15(2), 173-183. https://doi.org/10.32985/ijeces.15.2.6

[21] Bunker, R., & Susnjak, T. (2022). The application of machine learning techniques for predicting match results in team sport: A review. Journal of Artificial Intelligence Research, 73(3), 1285-1322. https://doi.org/10.1613/jair.1.13509

[22] Teixeira, J. E., Afonso, P., Schneider, A., Branquinho, L., Maio, E., Ferraz, R., et al. (2025). Player tracking data and psychophysiological features associated with mental fatigue in U15, U17, and U19 male football players: A machine learning approach. Applied Sciences, 15(7), 3718-3730. https://doi.org/10.3390/app15073718

[23] Woltmann, L., Hartmann, C., Lehner, W., Rausch, P., & Ferger, K. (2023). Sensor-based jump detection and classification with machine learning in trampoline gymnastics. German journal of exercise and sport research, 53(2), 187-195. https://doi.org/10.1007/s12662-022-00866-3

[24] Sonalcan, H., Bilen, E., Ateş, B., & Seçkin, A. Ç. (2025). Action recognition in basketball with inertial measurement unit-supported vest. Sensors, 25(2), 563-275. https://doi.org/10.3390/s25020563

[25] Canbulat, O. A., Turgay, S. A. & Kara, E. S. (2025). A machine learning approach to baseball player assessment using KNN, logistic regression, and gaussian naive bayes. Financial Engineering, 3, 14-21. https://doi.org/10.37394/232032.2025.3.2

[26] Zhang, Y., Wang, X., Xiu, H., Ren, L., Han, Y., Ma, Y., Chen, W., Wei, G. & Ren, L. (2023). An optimization system for intent recognition based on an improved KNN algorithm with minimal feature set for powered knee prosthesis. Journal of Bionic Engineering, 20(6), 2619-2632. https://doi.org/10.1007/s42235-023-00419-w

[27] Cao, G., Zhang, Y., Zhang, H., Zhao, T., & Xia, C. (2024). A hybrid recognition method via Kelm with Cpso for Mmg-based upper-limb movements classification. Journal of Mechanics in Medicine and Biology, 24(06), 2350084. https://doi.org/10.1142/S0219519423500847