



Research article

Research on motion recognition based on multi-dimensional sensing data and deep learning algorithms

Jia-Gang Qiu, Yi Li, Hao-Qi Liu, Shuang Lin, Lei Pang *, Gang Sun and Ying-Zhe Song

Institute of Artificial Intelligence in Sports, Capital University of Physical Education and Sports, Beijing 100191, China

* **Correspondence:** Email: panglei@cupes.edu.cn.

Abstract: Motion recognition provides movement information for people with physical dysfunction, the elderly and motion-sensing games production, and is important for accurate recognition of human motion. We employed three classical machine learning algorithms and three deep learning algorithm models for motion recognition, namely Random Forests (RF), K-Nearest Neighbors (KNN) and Decision Tree (DT) and Dynamic Neural Network (DNN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Compared with the Inertial Measurement Unit (IMU) worn on seven parts of body. Overall, the difference in performance among the three classical machine learning algorithms in this study was insignificant. The RF algorithm model performed best, having achieved a recognition rate of 96.67%, followed by the KNN algorithm model with an optimal recognition rate of 95.31% and the DT algorithm with an optimal recognition rate of 94.85%. The performance difference among deep learning algorithm models was significant. The DNN algorithm model performed best, having achieved a recognition rate of 97.71%. Our study validated the feasibility of using multidimensional data for motion recognition and demonstrated that the optimal wearing part for distinguishing daily activities based on multidimensional sensing data was the waist. In terms of algorithms, deep learning algorithms based on multi-dimensional sensors performed better, and tree-structured models still have better performance in traditional machine learning algorithms. The results indicated that IMU combined with deep learning algorithms can effectively recognize actions and provided a promising basis for a wider range of applications in the field of motion recognition.

Keywords: motion recognition; IMU; Deep learning algorithm; Classical machine learning algorithm; daily actions

1. Introduction

Motion recognition has become increasingly important in many fields. Harrington, M E used sensors to recognize gestures, providing an alternative method for computer input for patients with motor disorders and other conditions [1]. In [2], game interaction is carried out by recognizing human actions to improve game realism. Energy consumption and exercise intensity can be estimated by identifying action patterns in physical activities. In [3], wearable sensors are used to monitor the load of badminton training. In [4], wearable sensors are used to identify the type of movement and then estimate the intensity. However, human actions are complex and variable. Many researchers are committed to improving the recognition rate of motion.

At present, there is a growing amount of research on motion recognition using IMU sensors, because IMU costs less [5] and algorithms can be used to build models for recognizing action intentions [6]. However, the recognition rate of sensor-based motion recognition is influenced by a number of factors Cui, JW et al. wore IMU on the upper limb to identify the intention of upper limb movements [7], while Sarker, A monitored the movements of the upper body and looked for the optimal location for sensor placement [8]. Research shows that IMU sensors worn on the waist are not sensitive to interference in motion recognition [9], but the recognition performance depends on the specific recognition motion. Although the combination of multiple IMU sensors can improve the recognition rate of different actions, too many of them can also lead to over-fitting [10]. After collecting data through IMU sensors, it is necessary to extract features in order to facilitate the classification of the final algorithm. Zhuang W et al. extracted time-domain features and frequency-domain features from human motion data and used support vector machines to recognize actions such as walking and running. The results showed that the scheme can recognize body actions [11]. Li Q et al. extracted 60 features in the time-domain and frequency-domain. They classified 21 upper limb motions using a variety of classical machine learning algorithm models and argued that the number of features is correlated with the classifier [12]. Based on the extracted multi-source features, machine learning algorithms are often used to model and analyze IMU data to achieve accurate classification and recognition of different actions. Zhao Y et al. used an IMUs-framed microelectromechanical system to collect data on human actions during sitting, standing, running, walking up and down stairs, etc. They used a two-stage classification algorithm based on DT for classification and achieved a recognition rate of 95% [13].

In recent years, deep learning algorithms have developed rapidly and are gradually being applied in various fields. Some academics have used deep learning algorithmic models to process IMU data and have found that using these algorithms can improve the recognition rate of some actions partly. For example, [14] uses four deep learning models based on the three-axis accelerometer to identify seven basic actions. In [15], four deep learning models were constructed based on three-axis accelerometers and overlapping windows to identify three kinds of physical activities. Qi W et al. used a deep convolutional neural network model combined with IMU to recognize 12 daily actions and achieved a recognition rate of 93.89%, which is higher than that of classical machine learning algorithm models [16]. Vu H T T et al. used three deep learning algorithm models and a RF model combined with IMU to recognize four actions such as walking horizontally and going up and down stairs, and found that CNN and Long Short-Term Memory (LSTM) outperformed the other models [17]. In related research, some studies only focus on a few data dimensions, some use a single algorithm model and there are few studies on the seven parts of the body. Therefore, this study will integrate basic information and action information for motion recognition and use multiple algorithms to compare the performance of IMU wearing different parts.

Some existing studies combine with deep learning algorithm models or wear more sensors to improve the recognition performance. The objective of this study is to reduce the discomfort of the wearer by comparing the effects of different configurations of motion recognition to reduce sensor data during motion. The main contents of this study are as follows: 1. We collected information on 13 daily human actions and construct a multi-dimensional dataset based on basic human information (height, weight, age and gender) and IMU data. 2. We preprocessed and extracted features from the dataset. 3. We used various classical machine learning algorithms and deep learning algorithms to construct recognition models to identify and classify daily actions. 4. We compared and analyzed the impact of seven IMU sensor wearing positions (i.e., left upper arm, right upper arm, waist, left thigh, right thigh, left calf and right calf) on daily motion recognition.

2. Materials and methods

2.1. Participant's information

This experiment was completed in the laboratory of Capital University of Physical Education and Sports. The Ethics Committee at the Capital University of Physical Education and Sport approved the study protocol. A total of 26 students (aged 20-30 years old) were recruited as participants through a combination of online and offline methods with a male to female ratio of 1:1. Basic information of participants is shown in Table 1. All participants had no contraindications for exercise (such as hypertension, cardiovascular diseases and other diseases that are not suitable for exercise), no lower limb joint injury (open and closed), no cardiovascular diseases, no skin allergy, no hernia and other contraindications in the last 3 months. All participants were informed of the study prior to the experiment and signed subject informed consent.

Table 1. Basic information of participants.

characteristic	male	female
Height (cm)	180.81 ± 7.37	165.46 ± 5.71
Weight (Kg)	74.22 ± 6.07	59.82 ± 8.84
Age (years)	22.08 ± 1.90	22.92 ± 2.09
Number	13	13
BMI (kg/m ²)	22.75 ± 1.97	21.79 ± 2.52

2.2. Experimental equipment

The h/p/cosmos para treadmill (170&190/65 (SB), h/p/cosmos) can debug different speeds through the COM3 serial port, making the running process relatively stable and effectively controlling speed. The Polar heart rate belt (H10, Polar sports tester) collects ECG signals in real time through sensors worn on the human chest. The heart rate data was measured by recognizing the ECG signals, allowing real-time observation of heart rate data on the upper computer. Wit-Motion can be adjusted in the frequency range of 1–1000Hz. It collects the acceleration, angular velocity and Euler Angle signals of moving objects and calculates quaternion. Finally, the relevant data is transmitted to the upper computer through Bluetooth 5.0. The height and weight measuring instrument can accurately measure the participant's height and weight.

2.3. Experimental design

The participants did not engage in too much physical activity within 24 hours before the experiment, and conducted the experiment at least 1 hour after eating. The laboratory is kept comfortable and quiet, with a room temperature between 20–30°C. Before officially starting the test, the whole test process and precautions have been made clear to the participants. The main equipment in this experiment included the h/p/cosmos para treadmill, the Polar heart rate belt, Wit-Motion (BWT901BCL5.0) and height and weight measuring instrument. To reduce errors, the Wit-Motion IMU is uniformly worn with the data transmission port flag facing upwards and sets the acquisition frequency to 100Hz. This study collected 13 body actions that people often perform in their daily lives, as shown in Table 2.

Table 2. 13 body actions in daily activities.

Num	Action	Explanation	Label
1	lying	Lying on the Yoga mat naturally	1
2	sitting	Sitting in a chair naturally	2
3	standing	Standing naturally	3
4	stand-to-squat	Stand steadily to squat	4
5	squat-to-stand	Squat steadily to stand	5
6	stand-to-lie	Stand steadily to lie	6
7	lie-to-stand	Lie steadily to stand	7
8	walking	Walk on Treadmill at a speed of 4km/h	8
9	upsloping	Walk on the Treadmill at a speed of 5Km/h and a gradient of 10%	9
10	running	Run on Treadmill at a speed of 7km/h	10
11	cycling	The power is M: $0.98 * \text{body weight}$, F: $0.784 * \text{body weight}$, on a power bike	11
12	upstairs	Up the stairs in the stairwell	12
13	downstairs	Down the stairs in the stairwell	0

After starting the experiment, the participant's basic information (height, weight, age and gender) was collected first. Then the participant wore a polar heart rate belt and Wit-Motion Inertial Measurement Unit, as shown in Figure 1. After that, the participant was instructed to sit still for five minutes. Start the lying first. After the participant has been lying down steadily, start timing for four minutes and perform transitional actions (stand-to-squat and squat-to-stand) for eight minutes. After completion, rest until the heart rate is below 80 beats/min. The above scheme was repeated for each participant. Participant's motion data were recorded at the end of the test.



Figure 1. The position of the sensor during the experiment.

2.4. Data preprocessing and feature extraction

We collected acceleration data and angular velocity data in body actions, and the data of each action were collected for 4 minutes. To improve data accuracy, delete the data in the first 40s and leave three minutes and 20 seconds. Since the human limb frequency in daily life is usually between 0 and 20Hz, a Butterworth low-pass filter was used to filter out high-frequency noise in acceleration and angular velocity, and the cut-off frequency was set to 20Hz.

Feature extraction is a prerequisite for algorithm model to have better recognition results. Representative features can effectively improve the model's performance. In [18], the time-domain and frequency-domain features in IMU data were extracted to identify the three types of actions and achieved superior results. This study used a sliding window to extract time-domain and frequency-domain features from six-axis sensing data (acceleration and angular velocity). The length of the

sliding window is 10.24s and the overlap window is 50%. In [19], 142 features were extracted for seven types of action recognition. In [20], mobile phone sensor data features were extracted for human activity recognition. The features selected for each axis extraction in this study are shown in Table 3 and 4. A total of 192 features are extracted from the six-axis, and combined with basic information features (height, weight, age and gender), forming 196 features constituting a single sample. Then the corresponding labels were set for the sample according to the action category. We combined the samples into a dataset, removed the samples which had missing values and infinite series and used the above dataset as the basis for subsequent research for training and validation.

Table 3. Time-domain feature.

Features	Equation	Features	Equation
mean value	$F_1 = \frac{1}{N} \sum_{i=1}^N x_i$	amplitude	$F_{10} = (MAX(x) - MIN(x))/2$
median	$F_2 = L + (\frac{n}{2} - c)/f$	peakedness	$F_{11} = \frac{1}{N} \sum_{i=1}^N (\frac{x_i - \bar{x}}{F_5})^4$
maximum	$F_3 = MAX(x)$	skewness	$F_{12} = \frac{1}{N} \sum_{i=1}^N (\frac{x_i - \bar{x}}{F_5})^3$
minimum	$F_4 = MIN(x)$	average rectified value	$F_{13} = \frac{1}{N} \sum_{i=1}^N x_i $
standard deviation	$F_5 = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$	crest factor	$F_{14} = \frac{X_p}{X_{rms}}$
range	$F_6 = MAX(x) - MIN(x)$	clearance factor	$F_{15} = \frac{X_p}{X_r}$
number of zero crossing points	$F_7 = \sum_{i=1}^{N-1} sign(x_i) - sign(x_{i+1}) $	form factor	$F_{16} = \frac{X_{rms}}{ \bar{X} }$
square amplitude	$F_8 = (\frac{1}{N} \sum_{i=1}^N \sqrt{ x_i })^2$	pulse index	$F_{17} = \frac{MAX(x)}{ \bar{X} }$
mean square root	$F_9 = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$	quartile difference	$F_{18} = Q3 - Q1$

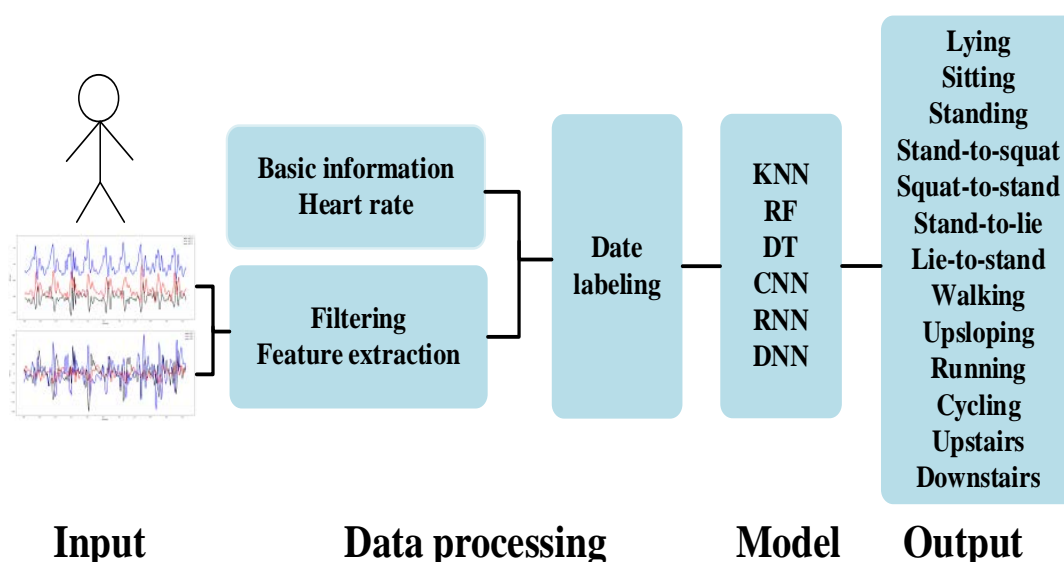
2.5. Algorithm model

In this study, three classical machine learning algorithms (KNN, RF and DT) and three deep learning algorithm models (RNN, CNN and DNN) were used to classify daily body actions. This section will introduce the above algorithms in detail, using the same dataset to evaluate the highest performance algorithm model and the identification process is shown in Figure 2.

Table 4. Frequency-domain feature.

Features	Equation	Features	Equation
mean value	$F_1 = \frac{1}{K} \sum_{k=1}^K s_k$	average rectified value	$F_8 = \frac{1}{K} \sum_{k=1}^K s_k $
variance	$F_2 = \frac{\sum_{k=1}^K (f_k - s_2)^2 s_k}{\sum_{k=1}^K s_k}$	skewness	$F_9 = \sqrt{\frac{\sum_{k=1}^K (s_k - \bar{s})^4}{F_2^4}}$
total energy	$F_3 = \sum_{k=1}^K s_k$	amplitude factor	$F_{10} = \frac{ Y(f) }{ X(f) }$
maximum	$F_4 = MAX(s)$	peakedness	$F_{11} = \frac{\frac{1}{K} \sum_{k=1}^K (f_i - \bar{f})^3}{(\frac{1}{K} \sum_{k=1}^K (f_i - \bar{f})^2)^{3/2}}$
minimum	$F_5 = MIN(s)$	gravity frequency	$F_{12} = \frac{\sum_{k=1}^K f_k s_k}{\sum_{k=1}^K s_k}$
standard deviation	$F_6 = \sqrt{F_2}$	mean square frequency	$F_{13} = \frac{\sum_{k=1}^K f_k^2 s_k}{\sum_{k=1}^K s_k}$
range	$F_7 = MAX(s) - MIN(s)$	root mean square frequency	$F_{14} = \sqrt{\frac{\sum_{k=1}^K (f_k - s_2)^2 s_k}{\sum_{k=1}^K s_k}}$

F_n in the table represents the corresponding result, other equations with F_n represent the use of that result, where $n = 1, 2, 3 \dots n$.

**Figure 2.** Data processing and model validation flow chart.

2.5.1. Classical machine learning algorithms

In this study, we construct classification models using three classical machine learning algorithms named KNN, DT and RF. KNN is a supervised learning algorithm with a relatively simple structure

which is often used in classification and regression scenarios. The KNN algorithm constructs an N-dimensional feature space from an N-dimensional training set and calculates the position of each point in the feature space. Then, put the test samples into the feature space and calculate the Euclidean distance between the test samples and the training samples [21]. Determine the label with the highest number of k-th training samples closest to the test sample and assign this test sample to the above label, the expression is as follows:

$$d(x_0, x_i) = \sqrt{\sum_{j=1}^n (x_{0j} - x_{ij})^2} \quad (2.1)$$

Therefore, the value of k is an important factor affecting classification. This study used a grid search method to obtain the best comprehensive performance of the algorithm model when k=5 and used a 5-fold cross validation method to verify the model performance.

The principle of DT is relatively simple. DT is a machine learning algorithm constructed from trees. It follows the principle of minimizing the loss function when constructing models [22]. The model consists of a root node, a leaf node and an internal node. The root node contains all the samples. Each feature attribute is represented by the internal node and ultimately the decision result is represented by a leaf node. This study chose the gini training model. Since this study is not a large sample size, the criterion of criterion was chosen to divide the nodes while no constraint was placed on the maximum depth of the DT. The model performance was validated using a 5-fold cross validation method.

The RF algorithm is an integrated supervised machine learning algorithm that can identify large sample datasets. When a test sample enters the model, the DT in the forest determines which class the sample belongs to and selects one type of sample as the predicted value of the test sample through statistical method [23]. This study chose gini as the model training. Although there is a positive relationship between the number of trees and the accuracy, the accuracy will remain flat or fluctuate after its number increases to a certain level. Therefore, the number of trees in this study was set to 100, and samples were randomly extracted with replacement. The model performance was verified using a 5-fold cross validation method.

2.5.2. Deep neural network algorithms

In this study, we construct classification models using three deep neural network algorithms named CNN, RNN and DNN. CNN is a mathematical model that simulates the structure and capabilities of biological nerves. CNN has multiple neurons and through a variety of connections, it can form different neural networks. It has some simple judgment ability like living organisms [24]. The representative formula is as follows:

Convolution Formula:
$$Z_{i,j,k} = \sum_{p=0}^{f_h-1} \sum_{q=0}^{f_w-1} \sum_{c=1}^{C_{in}} W_{p,q,c,k} a_{i+p,j+q,c} \quad (2.2)$$

$Z_{i,j,k}$ represents the output of convolution, $W_{p,q,c,k}$ represents the weight of the convolutional kernel, $a_{i+p,j+q,c}$ represents the output of the c-th channel in the input layer at $(i+p, j+q)$ and f_h and f_w represent the height and width of the convolutional kernel, respectively. C_{in} represents the number of channels in the input layer.

Pooling formula:
$$a_{i,j,k} = \max_{p=0}^{p_h-1} \max_{q=0}^{p_w-1} a_{i \times s_h + p, j \times s_w + q, k} \quad (2.3)$$

$a_{i,j,k}$ represents the output of pooling layer, p_h and p_w represents the height and width of the pooling core, respectively, and s_h and s_w represents the step size of the pooling kernel, respectively.

Figure 3 shows the 1D CNN network applied in this study. This study used three layers 1D convolutions layers (Conv1d) with a kernel size of 5 and a step size of 1. The first and second pooling layer size is set to 2 and the last pooling layer size is set to 45. Two fully connected layers are used. The number of neurons is 64 and 32, respectively, the activation function is tanh and finally the softmax activation function is used for output. Adam optimizer was used for optimization. Set the learning rate to 0.01, batch to 1000 and epoch to 115.

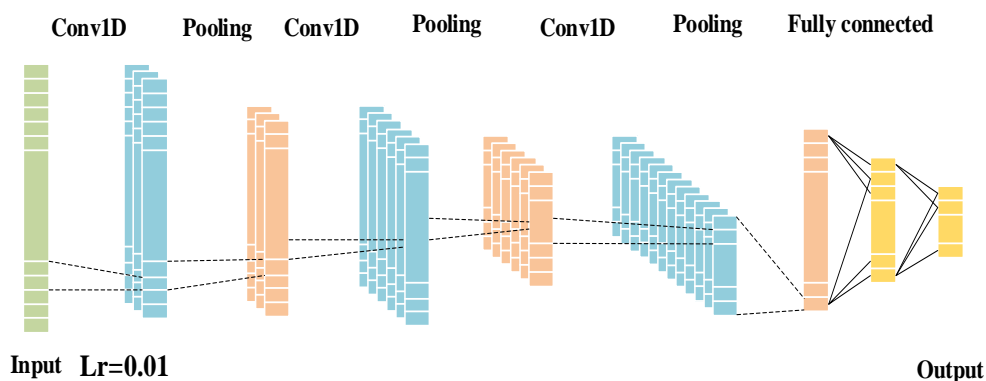


Figure 3. 1D CNN network structure diagram.

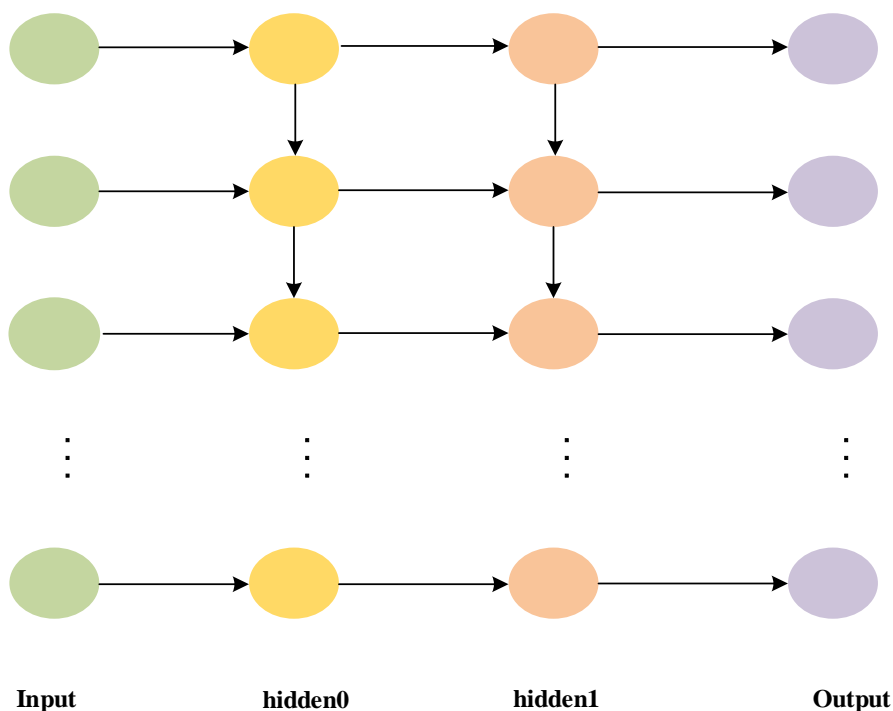


Figure 4. RNN network structure diagram.

RNN is capable of processing sequential data and has neurons with self-feedback, memory capability, parameter sharing and other characteristics [25]. These capabilities make it particularly useful for processing non-linear data. This study has constructed an RNN network structure that includes an input layer, two hidden layers and an output layer. The number of neurons is 50. Each batch has 1000 samples, the activation function is tanh, the learning rate is 0.02, the optimizer is Adam and 115 epochs are set to complete the training. The implementation principle is shown in Figure 4.

DNN is an unsupervised learning algorithm constructed from multi-layer networks which is a neural network composed only of input layers, hidden layers and output layers [26]. It includes forward propagation, back propagation and activation function. Its representative formula is as follows:

$$\text{Forward propagation:} \quad z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]} \quad (2.4)$$

$z^{[l]}$ represents the input of layer I, $W^{[l]}$ represents the weight of layer I, $a^{[l-1]}$ represents the input of layer I-1 and $b^{[l]}$ represents the bias of layer I.

$$\text{Activation function:} \quad a^{[l]} = g(z^{[l]}) \quad (2.5)$$

g represents the activation function.

$$\text{Back propagation:} \quad \frac{\partial J}{\partial z^{[l]}} = \frac{\partial J}{\partial a^{[l]}} \odot g'(z^{[l]}) \quad (2.6)$$

J represents the loss function, \odot represents the product at the element level and g' represents the derivation of activation function.

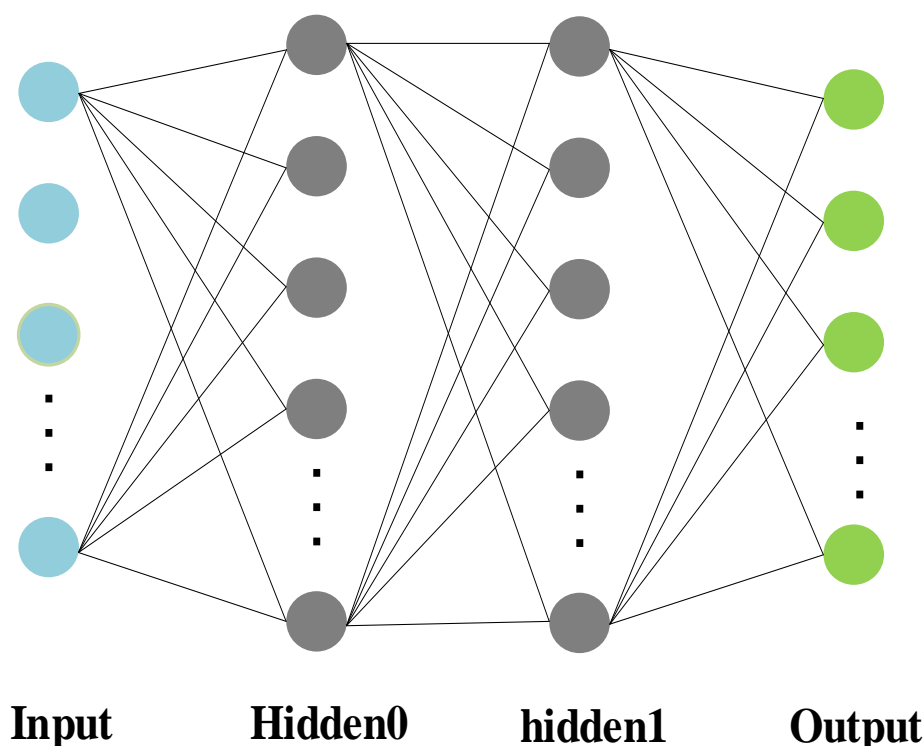


Figure 5. DNN network structure diagram.

The principle is shown in Figure 5. This study constructs a 2-layer hidden layer network structure, the number of neurons is 128 and 64, respectively, using tanh as an activation function. Model training

was conducted with 500 batches at a time, and a total of 180 epoch trainings were performed. Adam was employed as the optimizer for the network structure, and the category prediction was done using the softmax activation function. We set the learning rate to 0.01.

2.5.3. Model operation and evaluation

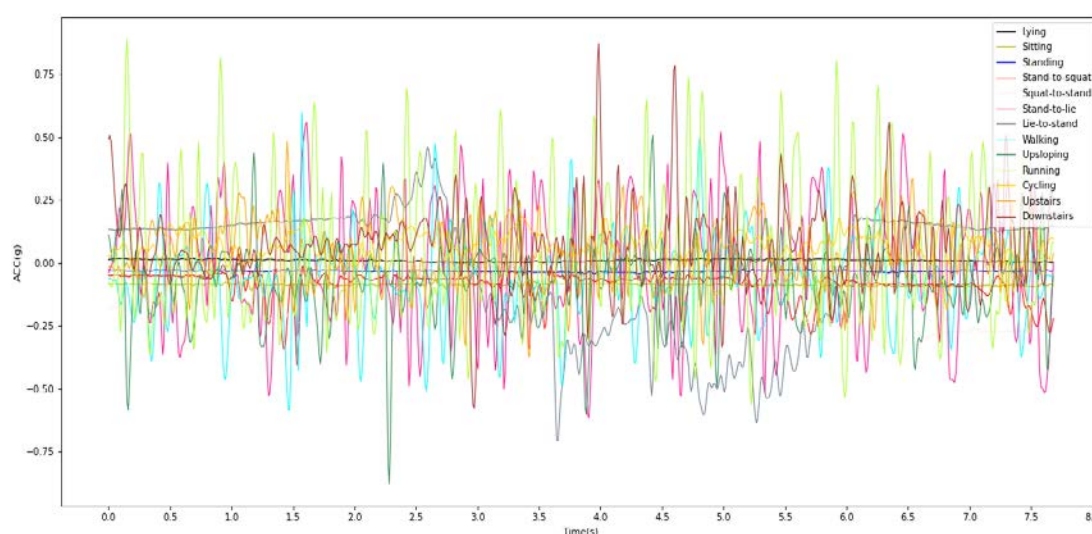
To evaluate the performance of the model, a cross-validation approach was utilized in this study. The data set samples were randomly divided into two parts, the training set and the test set, with the ratio set at 4:1. The training set was used for model training and the test set was used to test the performance of the model. Traditional machine learning algorithms judge model performance by model accuracy index. However, in the case of deep learning models, both recognition accuracy and loss functions are taken into account.

In this study, python3.9 was selected and compiled it using PyCharm Community Edition 2020.3.3x64 on Windows 11.

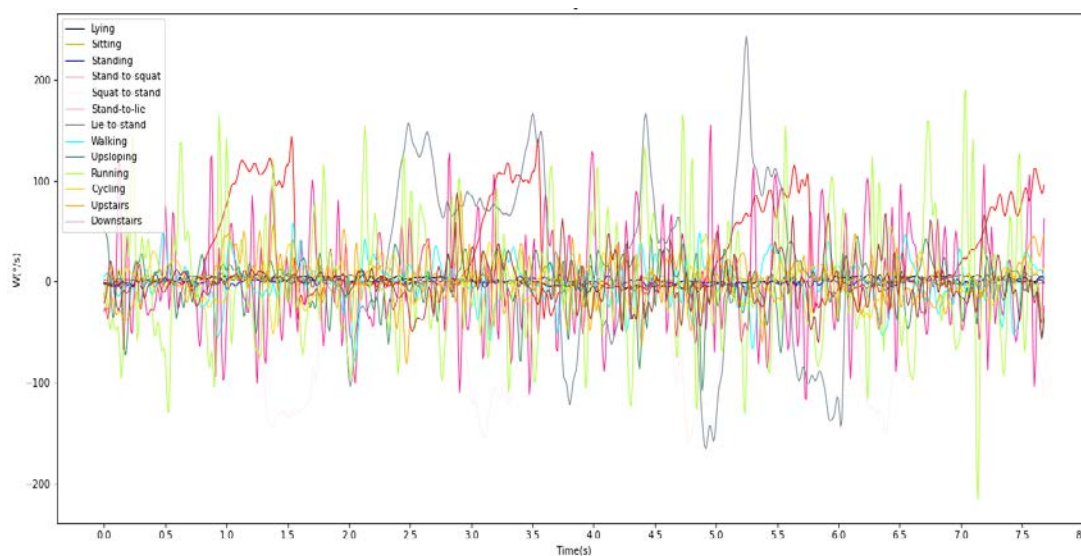
3. Results analysis and discussion

3.1. IMU data analysis

The analysis of the waist IMU data in 13 actions is used as an example to introduce the subsequent data processing steps. The curves of acceleration and angular velocity in the x-axis direction versus time in the waist inertial sensor are given respectively, as shown in Figure 6. The figure clearly exhibits the IMU changes for different actions. Although the trends in acceleration and angular velocity are almost identical for the stationary actions (lying, sitting, standing, etc.), there are still some differences in amplitude. The varying non-stationary actions show significant changes in acceleration and angular velocity as they proceed during activities. Based on the data presented, it can be inferred that different actions will have different characteristics in IMU motion data and IMU is a data generated in time series. More extensive data analysis is necessary due to variations in the characteristics of the IMU data across different actions over time. Furthermore, it is difficult to directly distinguish different actions through IMU data.



(a)



(b)

Figure 6. Data variation curve of waist IMU in 13 actions: (a) acceleration in x-axis (b) angular velocity in x-axis.

3.2. Comparison of motion recognition of different parts based on classical machine learning algorithms

Based on the 13 types of actions that participants often perform in daily life, this study collected motion data from IMU of different body parts, and constructed multiple machine learning algorithm models to accurately distinguish actions. In this section, the evaluation scheme proposed in Section 2.4.3 will be used for model evaluation. The data set will be divided into training set (80%) and test set (20%). The training set is used to build an algorithm model, and the test set is used to evaluate model performance.

To construct the models, this study employed three classical machine learning algorithms, namely KNN, DT and RF. After training the model on the training set, the test set was put into the model for testing. The accuracy of three classic machine learning algorithms is shown in Table 5. The accuracy of the training set was generally higher than that of the test set. The accuracy of the training set of DT and RF algorithms is 100%, while the accuracy of KNN unable to reach. Comparing the test set accuracy of the three algorithms, the overall accuracy of the RF algorithm for different action recognition is higher than that of KNN and DT algorithm, the recognition accuracy of different parts is basically above 90%, with a maximum of 97%. Both the KNN and RF algorithms demonstrate similar accuracy rates when modeling based on IMU data from different parts on the body.

Among the seven parts of the body, the waist has the highest accuracy with all three algorithmic models performing above 94%. The accuracy rates of left and right limbs are similar albeit slightly higher for the left limb. In comparison to the upper limbs, the lower limbs exhibited higher recognition accuracy and relatively stable performance for all 13 whole body actions. The above results suggested that it is feasible to distinguish different actions based on traditional machine learning algorithms.

Table 5. Results of 13 types of action recognition based on machine learning algorithms.

	KNN		DT		RF	
	train	test	train	test	train	test
Left_arm	92.47%	88.79%	100%	88.30%	100%	91.02%
Right_arm	92.25%	87.47%	100%	87.47%	100%	89.47%
waist	96.71%	95.31%	100%	94.85%	100%	96.67%
Left_thigh	96.49%	93.47%	100%	94.03%	100%	94.41%
Right_thigh	95.23%	92.69%	100%	93.14%	100%	94.91%
Left_calf	95.57%	93.07%	100%	93.78%	100%	94.70%
Right_calf	94.81%	91.95%	100%	93.40%	100%	93.48%

3.3. Building motion recognition models for different parts based on deep learning algorithms

Following the method described in Section 2.4.2, this study built a neural network model and utilized three deep learning algorithms (RNN, CNN and DNN) to classify actions. The same training set was first put into the model for training, and then the testing set was put into the model for testing. The accuracy of the training and testing sets recognized by the three models in seven parts is shown in Table 6. The results showed that the RNN had excellent performance on the training set but poor performance on the test set, while the CNN and DNN models showed outstanding performance on both the testing and training sets. After comparing the recognition performance of the three deep learning models by testing set results, it was observed that DNN showed the highest accuracy rate, followed by CNN. At the same time, the results obtained from DNN and CNN models surpassed those of the KNN, DT and RF algorithms, demonstrating that deep learning networks can recognize hidden features of different actions for differentiation. The RNN model performed the worst, sometimes worse than the three traditional machine learning algorithms. This could be attributed to the fact that the time sequence characteristics of IMU data in this study have not been fully applied.

Table 6. Results of 13 types of action recognition based on deep learning algorithms.

	RNN		CNN		DNN	
	train	test	train	test	train	test
Left_arm	98.24%	87.35%	97.14%	92.30%	99.31%	96.29%
Right_arm	96.70%	83.76%	98.44%	91.47%	99.48%	94.84%
Waist	99.99%	89.34%	98.01%	96.72%	99.99%	97.71%
Left_thigh	96.68%	87.92%	97.59%	94.93%	99.93%	97.65%
Right_thigh	96.89%	88.01%	97.42%	92.98%	99.92%	96.39%
Left_calf	99.99%	91.66%	98.91%	94.74%	100%	97.71%
Right_calf	100%	90.02%	97.46%	94.21%	99.81%	96.74%

Both DNN and CNN models demonstrated stable performance while recognizing different actions across body parts, with accuracy rates consistently above 90%. Consistent with machine learning algorithm model recognition results, recognition based on sensing data wore on the waist had the highest accuracy. The DNN model showed a recognition rate of over 95%. In addition, the accuracy

of the left limb was better than that of the right limb. On the whole, in addition to the RNN model, the deep learning algorithm models have better recognition effects for different actions than traditional machine learning algorithms, which can reduce recognition errors caused by different wearing locations.

To conduct a more comprehensive analysis of the CNN and DNN models' recognition capabilities for the 13 actions, waist sensing data with the best effect was selected for analysis. The Loss and Accurate curves of the CNN and DNN model algorithms for training models on waist data are shown in Figure 7. The loss curves of both models gradually tend to zero over time, indicating the effectiveness of feature selection for the dataset. The DNN model showed a more rapid decrease in loss curve than the CNN model, alongside a quicker increase in recognition rate. Both of them achieved better performance at 60 epochs. At the same time, the two models exhibited an overfitting phenomenon after 60 epochs. But the DNN can recover quickly and the Loss can further tend to zero.

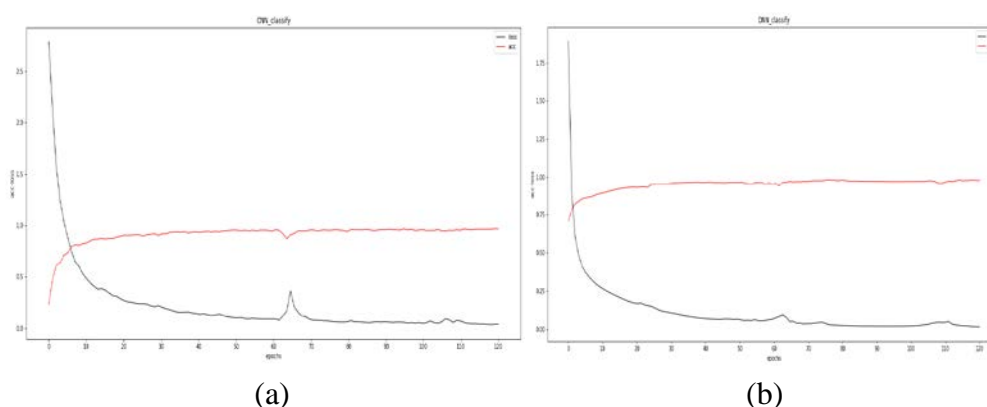


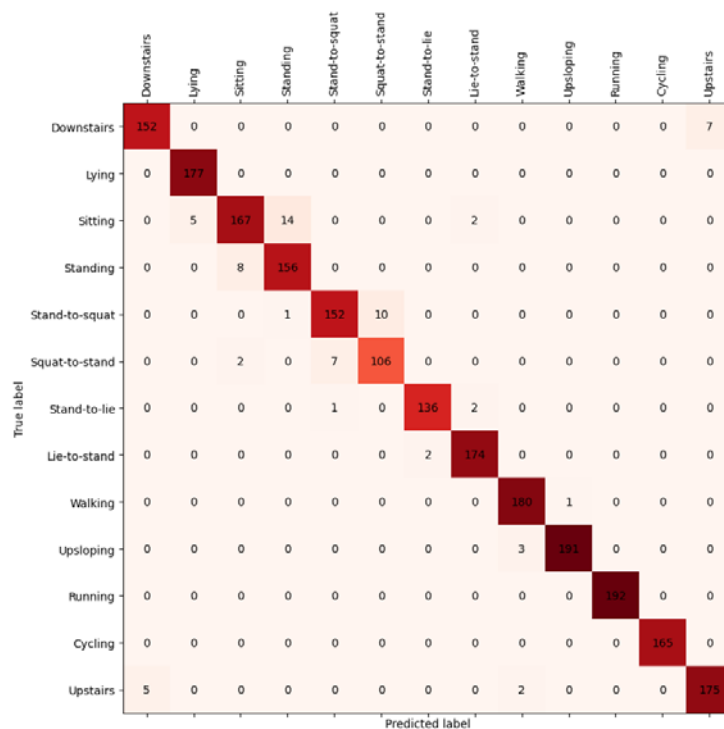
Figure 7. Accuracy and Loss curves of CNN and DNN models for waist data: (a) CNN algorithm model (b) DNN algorithm model.

In this study, a classification confusion matrix was constructed to deeply compare the recognition effects of CNN and DNN networks on different actions. According to the confusion matrix of the two algorithms (as shown in Figure 8), the error recognition rate of standing, sitting, walking up the stairs, walking down the stairs, standing to squat and squatting to the stand is high. The high error recognition rate for standing and sitting actions could be attributed to the waist posture being nearly identical in both actions. For the other actions, the sliding window used during timeslice could be responsible for the misclassification of the first half of the action and the second half of the action into the same category, consequently leading to incorrect classification.

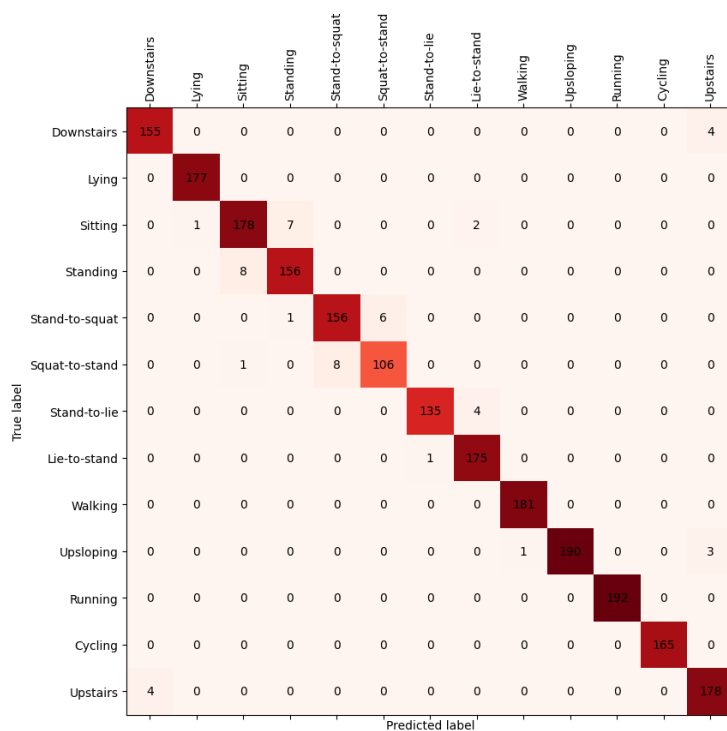
3.4. Discussion

This study tested the performance of multiple algorithmic models in motion recognition. The recognition results of the testing set of six algorithms are compared as shown in Figure 9. According to the results, the recognition performance of KNN, DT and RF algorithms was relatively close. The performance of deep learning algorithms, including RNN, CNN and DNN, exhibited a considerable variation with RNN having the lowest recognition rate and only 89% at the waist. This result could be attributed to the RNN's sensitivity towards time-series data. RNN is related not only to current inputs but also to previous and future outputs. Therefore, the performance of model is slightly worse in some transition actions or scenarios with opposite actions. The DNN algorithm performed better than other algorithms with an optimal recognition rate of 98%. Moreover, the model's structure is relatively

simple, and it can efficiently obtain output. By comparing classical machine learning algorithms and deep learning algorithm models, it was found that RF showed good performance in classical machine learning algorithms with an average accuracy of 93% and DNN algorithms with an average accuracy of 96.8%. This result highlighted the deep learning algorithm model's superior recognition capabilities in this experiment.



(a)



(b)

Figure 8. Waist confusion matrix: (a) CNN algorithm model (b) DNN algorithm model.

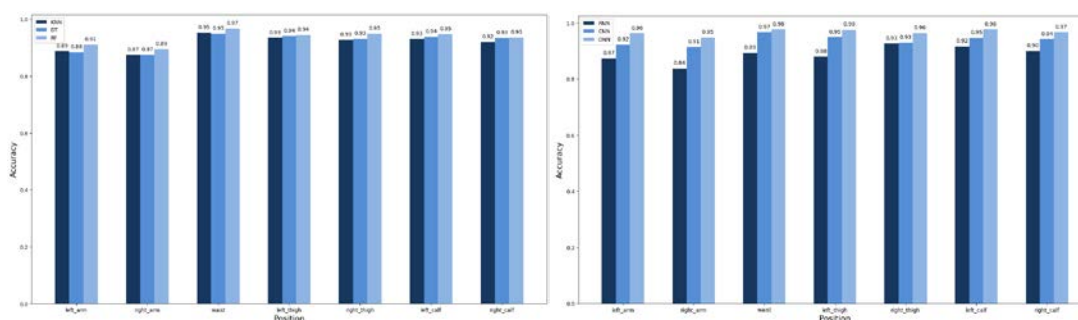


Figure 9. The comparison of the accuracy of traditional machine learning algorithms and deep learning algorithms for the recognition of different actions (a) traditional machine learning algorithms (b) deep learning algorithms.

4. Conclusion

Since motion recognition technology finds extensive applications in fields such as medicine, sports and military, there is an increased drive to achieve better recognition performance for such devices. Therefore, this study proposed using multi-dimensional data to identify 13 daily actions and collected six-axis inertial sensor data from seven parts of the body (left upper arm, right upper arm, waist, left thigh, right thigh, left calf and right calf) of 26 participants in real time. We trained various algorithm models and verified them. This scheme can be implemented at a lower cost, avoiding risk factors such as privacy discovery. Based on the experimental results, it is feasible to extract features from the time-domain and frequency-domain and perform motion recognition. To explore the optimal location for wearing inertial sensors, this study attached six-axis sensors on seven body parts of participants and concluded that the waist had the best recognition effect, followed by the thigh part, while the upper limbs exhibited comparatively lower recognition efficacy. The above recognition rates have reached over 80%. The findings of this study offer a theoretical underpinning and feasibility support for multi-sensor motion recognition. It is noteworthy that the time-domain and frequency-domain features extracted in the study are relatively simple because these are commonly used features. In the future, we will try to overcome these difficulties and use other methods to extract more effective features to improve the recognition rate or save time and cost.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is partially supported by the Beijing Science and Technology Plan Project (Grant No. Z221100005222031).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. M. E. Harrington, R. W. Daniel, P. J. Kyberd, A measurement system for the recognition of arm

- gestures using accelerometers, *Proceed. Institut. Mechan. Eng. Part H J. Eng. Med.*, **209** (1995), 129–134. https://doi.org/10.1243/pime_proc_1995_209_330_02
2. C. H. Chang, C. H. Yeh, C. C. Chang, Y. C. Lin, Interactive somatosensory games in rehabilitation training for older adults with mild cognitive impairment: Usability study, *JMIR Serious Games*, **10** (2022), e38465. <https://doi.org/10.2196/38465>
 3. T. H. Liu, W. H. Chen, Y. Shih, Y. C. Lin, C. Yu, T.Y. Shiang, Better position for the wearable sensor to monitor badminton sport training loads, *Sports Biomechan.*, (2021), 1–13. <https://doi.org/10.1080/14763141.2021.1875033>
 4. I. Pernek, G. Kurillo, G. Stiglic, R. Bajcsy, Recognizing the intensity of strength training exercises with wearable sensors, *J. Biomed. Inform.*, **58** (2015), 145–155. <https://doi.org/10.1016/j.jbi.2015.09.020>
 5. Y. F. Liu, Z. T. Li, L. H. Xiao, S. K. Zheng, P. C. Cai, H. F. Zhang, et al., FDO-Calibr: Visual-aided IMU calibration based on frequency-domain optimization, *Measurement Sci. Technol.*, **34** (2023). <https://doi.org/10.1088/1361-6501/acadfb>
 6. J. W. Cui, Z. G. Li, H. Du, B. Y. Yan, P. D. Lu, Recognition of upper limb action intention based on IMU, *Sensors*, **22** (2022), 1954. <https://doi.org/10.3390/s22051954>
 7. J. W. Cui, Z. G. Li, Prediction of upper limb action intention based on long short-term memory neural network, *Electronics*, **11** (2022), 1320. <https://doi.org/10.3390/electronics11091320>
 8. A. Sarker, D. R. Emenonye, A. Kelliher, T. Rikakis, R. M. Buehrer, A. T. Asbeck, Capturing upper body kinematics and localization with low-cost sensors for rehabilitation applications, *Sensors*, **22** (2022), 2300. <https://doi.org/10.3390/s22062300>
 9. S. T. Boerema, L. van Velsen, L. Schaake, T. M. Tonis, H. J. Hermens, Optimal Sensor Placement for Measuring Physical Activity with a 3D Accelerometer, *Sensors*, **14** (2014), 3188–3206. <https://doi.org/10.3390/s140203188>
 10. C. S. Xia, Y. Sugiura, Wearable accelerometer layout optimization for activity recognition based on swarm intelligence and user preference, *IEEE Access*, **9** (2021), 166906–166919. <https://doi.org/10.1109/access.2021.3134262>
 11. W. Zhuang, Y. Chen, J. Su, B. W. Wang, C. M. Gao, Design of human activity recognition algorithms based on a single wearable IMU sensor, *Int. J. Sensor Networks*, **30** (2019), 193–206. <https://doi.org/10.1504/ijsn.2019.100218>
 12. Q. Q. Li, Y. G. Liu, J. J. Zhu, Z. Chen, L. Liu, S. M. Yang, et al., Upper-limb motion recognition based on hybrid feature selection: algorithm development and validation, *JMIR Mhealth Uhealth*, **9** (2021), e24402. <https://doi.org/10.2196/24402>
 13. Y. J. Zhao, J. Zhao, L. Ding, C. C. Xie, Big data energy consumption monitoring technology of obese individuals based on MEMS sensor, *Wireless Commun. Mobile Comput.*, **2021** (2021), 1–11. <https://doi.org/10.1155/2021/4923804>
 14. V. Bijalwan, V. B. Semwal, V. Gupta, Wearable sensor-based pattern mining for human activity recognition: deep learning approach, *Industr. Robot Int. J. Robot. Res. Appl.*, **49** (2022), 21–33. <https://doi.org/10.1108/ir-09-2020-0187>
 15. M. Jaen-Vargas, K. M. Reyes Leiva, F. Fernandes, S. Barroso Goncalves, M. Tavares Silva, D. S. Lopes, et al., Effects of sliding window variation in the performance of acceleration-based human activity recognition using deep learning models, *PeerJ Computer Sci.*, **8** (2022), e1052. <https://doi.org/10.7717/peerj-cs.1052>
 16. W. Qi, N. Wang, H. Su, A. Aliverti, DCNN based human activity recognition framework with depth

- vision guiding, *Neurocomputing*, **486** (2022), 261–271. <https://doi.org/10.1016/j.neucom.2021.11.044>
17. H. T. T. Vu, H. L. Cao, D. B. Dong, T. Verstraten, J. Geeroms, B. Vanderborght, Comparison of machine learning and deep learning-based methods for locomotion mode recognition using a single inertial measurement unit, *Front. Neurobot.*, **16** (2022), 209. <https://doi.org/10.3389/fnbot.2022.923164>
 18. D. Hoareau, G. Jodin, P. A. Chantal, S. Bretin, J. Prioux, F. Razan, Synthetized inertial measurement units (IMUs) to evaluate the placement of wearable sensors on human body for motion recognition, *J. Eng.*, **2022** (2022), 536–543. <https://doi.org/10.1049/tje2.12137>
 19. A. Narayanan, T. Stewart, L. Mackay, A dual-accelerometer system for detecting human movement in a free-living environment, *Med. Sci. Sports Exercise*, **52** (2020), 252–258. <https://doi.org/10.1249/MSS.0000000000002107>
 20. W. H. Kong, L. L. He, H. L. Wang, Exploratory data analysis of human activity recognition based on smart phone, *IEEE Access*, **9** (2021), 73355–73364. <https://doi.org/10.1109/access.2021.3079434>
 21. H. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *Am. Statist.*, **46** (1992), 175–185. <https://doi.org/10.2307/2685209>
 22. J. R. Quinlan, Learning decision tree classifiers, *ACM Comput. Surveys*, **28** (1996), 71–72. <https://doi.org/10.1145/234313.234346>
 23. D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, Random forests for classification in ecology, *Ecology*, **88** (2007), 2783–2792. <https://doi.org/10.1890/07-0539.1>
 24. Z. W. Li, F. Liu, W. J. Yang, S. H. Peng, J. Zhou, A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, *IEEE Transact. Neural Networks Learn. Syst.*, **33** (2022), 6999–7019. <https://doi.org/10.1109/tnnls.2021.3084827>
 25. J. S. Han, E. G. Im, Implementation of Individual Gait Recognition using RNN, *KIISE Transact. Comput. Pract.*, **24** (2018), 358–362. <https://doi.org/10.5626/ktcp.2018.24.7.358>
 26. G. Montavon, W. Samek, K. R. Muller, Methods for interpreting and understanding deep neural networks, *Digital Signal Process.*, **73** (2018), 1–15. <https://doi.org/10.1016/j.dsp.2017.10.011>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)