

## Research Article

# Prediction of Behavior and Activity of Patients through Wearable Devices Based on Deep Learning

Decheng Zhang <sup>1</sup>, Hengjing Zhang <sup>2</sup>, and Chuanxin Zhao <sup>3</sup>

<sup>1</sup>School of Basic Courses, Bengbu Medical College, Bengbu, 233000 Anhui, China

<sup>2</sup>School of Software Engineering, University of Science and Technology of China, 215123 Suzhou, China

<sup>3</sup>School of Computer and Information, Anhui Normal University, Wuhu, 241000 Anhui, China

Correspondence should be addressed to Decheng Zhang; [zdc@bbmc.edu.cn](mailto:zdc@bbmc.edu.cn)

Received 4 June 2021; Revised 10 December 2021; Accepted 7 January 2022; Published 27 January 2022

Academic Editor: Manuel Alexandre

Copyright © 2022 Decheng Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the aging problem becoming more and more serious, muscle pain has become a common symptom. In order to help patients with rehabilitation training, it is necessary to monitor their activities in time. We propose a real-time monitoring method based on wearable devices. This method uses a wireless body area network for health care. Specifically, in the first step, we developed a wearable device based on ZigBee with low cost and low weight. Secondly, if only classifying the action at the current time, it will not meet the requirements of real-time monitoring. So, we design an end-to-end neural network model called ATCRNN to infer the actions to be made by users at the next time according to the data of the past few times. This model uses CNN and RNN to extract the spatial and temporal features of data and captures the context characteristics through self-attention. Finally, four volunteers wore equipment to participate in the experiment. The activity categories in the experiment are walking, sitting down, running, and climbing stairs. The accuracy of behavior inference reached 97% with ATCRNN. The compared results demonstrate that the performance of the proposed approach is superior to the result of other deep learning networks.

## 1. Introduction

The aging of population is a problem faced by many countries in the world. With the prolongation of human life span and the declining fertility rate, the proportion of the global elderly population is increasing, which brings new challenges to the existing health care system, mainly including the problems of limited medical resources and shortage of medical personnel. With the development of microcomputer technology, intelligent mobile devices and various wearable sensors are widely used. In recent years, mobile devices and sensors have made significant progress in size, cost, and power consumption, which provides a new data source for studying people's daily activities and behaviors. As a revolutionary technology, wearable technology plays an important role in health care, sports health, smart city, military operations, and other fields. Based on the rapid development of wearable technology, human activity recognition has been paid more and more attention [1, 2].

Wireless body area network (WBAN) [3] is a kind of wireless sensor network on the human body. It can collect the life information and posture information of human body in real time and transmit data to the host computer by selecting the appropriate position on the human body. At present, most of the human gesture recognition technologies are based on video capture technology, which requires high-precision cameras to monitor the human body in real time. Therefore, there are some disadvantages such as high investment, visual shadow, and user privacy. The emergence of wireless body area network provides a new idea for human posture recognition in medical treatment [4]. Compared with video recognition, wearable devices have good convenience.

This paper has carried out the following work:

- (1) In order to detect the real-time activity of user, a wearable device based on ZigBee is designed in this paper. This wearable device is small and easy to carry. Patients can tie the equipment to their waist

and move. The wearable devices transmit the data to the server, and the server recognizes the human activity in real time

- (2) Because the data collected by wearable devices is time series, it cannot be directly applied to deep learning networks. In this paper, we propose a method based on sliding window to separate the data sequence
- (3) In order to recognize human activities better, this paper analyzes the temporal and spatial characteristics of human activities; this paper proposes a time series convolution network model based on self-attention. The model has good recognition effect. In the experiment, the accuracy of behavior prediction is 97%

## 2. Related Works

Recently, human gesture recognition has attracted many researchers' interests. Some methods like  $k$ -Nearest Neighbours (kNN) [5], Support Vector Machine (SVM) [6], and random forest [7] have been used in human gesture recognition. In [8], a multisensor multiclassifier hierarchical fusion model is proposed in which based on entropy weight using wearable inertial sensors for human activity recognition, the recognition accuracy rate is high. In addition, LSTM, CNN, deep feed-forward networks, and CNN-LSTM are used in human gesture recognition. In [9], a convolutional neural network (CNN) is proposed for a real-time human activity classification method. In [10], a tensor approach with a novel deep learning architecture is proposed for multivariate time series classification in human gesture recognition. In [11], the authors use CNN for extracting local features from accelerometer data. In [12], a deep convolutional neural network is designed by exploiting the inherent characteristics from smartphone sensors to obtain effective human gesture recognition.

The multilevel hierarchical algorithm can identify the human posture more accurately according to the different characteristics of different human posture. A gesture recognition algorithm based on SOM (self-organizing map) classifier and Hebbian network can successfully recognize 24 kinds of American sign languages, and the recognition rate is 97.1%. However, the video recorder is used in the hardware system, which has defects in protecting the privacy of each person [13]. In [14], the pressure sensor and acceleration are made into intelligent insoles and placed on the sole of the foot. The human body posture is recognized by using SVM (signal vector magnitude) and MLP (multi-layer perception) as classifiers. The different types of sensors and the lower position of sensors affect the attitude recognition rate. In [15], a kind of glove was made from a KPF displacement sensor to identify the state of fingers. Taking displacement as the only parameter to identify posture has a great contribution to the real-time monitoring of stroke patients' gesture. However, the biggest defect of the device is that it affects the user's normal activities. Reference [16] evaluated and compared the accuracy of bending sensor in measuring the bending angle of human knee and proposed that the

bending sensor can identify walking, running, cycling, and other movements by collecting the bending degree of knee. In Reference [17], an attitude recognition system based on multilevel hierarchical algorithm is designed. The human posture is divided into simple posture and complex posture, and the recognition rate is 82.87%. In [18], two axis acceleration sensors were placed on the back and head of human body for posture recognition. The algorithm can recognize the stand, walk, and run attitude and has the characteristics of free movement, but its recognition rate is only 80%, the recognition rate is low, and the recognition attitude is less. CNN and Gru are frequently used methods [19]. Arshad et al. proposed a deep learning and fuzzy entropy-controlled skewness approach for human gait recognition [20]. Many studies have deal with HAR task from the perspective of deep learning and machine learning [21, 22]. However, in practical applications, a lot of time will be consumed in transmission, calculation, and feedback. All of these will result in time delay. If we need to reduce the time delay and provide real-time monitoring, it is far from enough to recognize the human activity just at the current time. Inferring the action at the next time in advance will be need to achieve real-time monitoring. Therefore, in this paper, we design a user behavior real-time monitoring method based on wireless body area network. Firstly, we design a wireless body area network data acquisition framework and develop a wearable device based on ZigBee. The device is embedded with MPU6050 6-axis acceleration gyroscope attitude sensor. The sensor is placed at the waist of the human body to collect the human body activity information. Then, the data will be transmitted to the host computer. Finally, this paper proposes an end-to-end neural network model called ATCRNN to infer the actions to be made by users at the next time according to the data of the past few times. It can accurately predict the walking, sitting, running, and climbing stairs, and the algorithm has good robustness and high recognition rate.

## 3. Wireless Body Area Network Data Acquisition Framework

*3.1. Overall Framework Design.* This paper designs a human activity data acquisition framework suitable for wireless body area network. The framework includes wearable sensing devices, intelligent computing terminals, and data cloud platform.

As shown in Figure 1, this paper designs a body posture data acquisition framework for wireless body area network. The framework includes wearable sensing devices, intelligent computing terminals, and data cloud platform. In the framework, wearable devices as sensing nodes are responsible for real-time acquisition of user's somatosensory signals, such as acceleration, angular velocity, and other inertial data. As a terminal node, intelligent computing terminal not only undertakes the function of communication but also undertakes the function of calculation. The intelligent computing terminal receives the data for calculation and sends it to the cloud for storage. The wearable device uses ZigBee as the wireless communication module and sends the collected

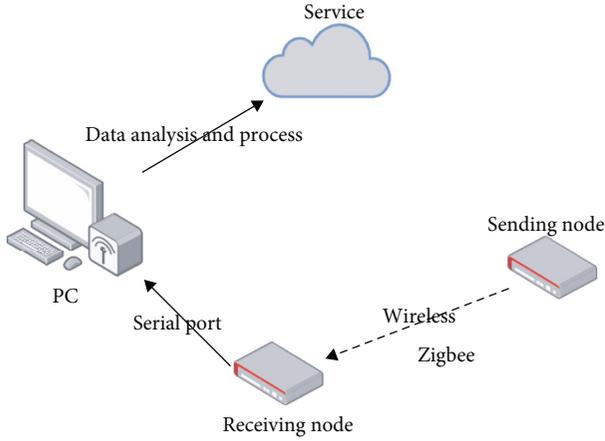


FIGURE 1: Overall framework.

data to the terminal node through ZigBee. As a visualization platform, data cloud platform obtains the data stored in the terminal node from the database and displays it in real time on the platform.

In the specific experimental process, the wearable device includes three components: STM32 is selected as the development board, MPU6050 is used as attitude sensor, and ZigBee module is used as communication module. ZigBee sink node sends data to the computer through serial port, and upper computer collects and calculates data. Finally, the results are displayed in the cloud platform.

**3.2. Data Collection and Experiment.** Based on the above framework system, we developed the upper computer software, which is responsible for real-time visualization of the collected data and saving it to the database. In order to verify the effectiveness of this method, simulation experiments are carried out. The host software is developed by python, PyQt is used as user interface component, and a friendly image interface is designed, which can display all kinds of perception data in real time. The collected data are shown in Figure 2.

In order to verify the effectiveness of the proposed method, a simulation scene experiment is carried out. Activity monitoring is an important issue in chronic disease management. We chose the most basic types of movements for monitoring. These actions include walking, sitting down, running, and climbing stairs. Bind the sensor node to the waist, as shown in Figure 3. In the experiments, two men and two women participated in the test. Table 1 shows their static information, including age, height, weight, and other basic physiological conditions. The subjects collected and labeled sensor data for four types of daily activities, including walking, sitting, running, and climbing stairs. One person's action at a time lasts at least 5 minutes. In addition, in order to ensure the unbiased results, the data are randomly selected and arranged in a random order.

## 4. Data Processing

The data collected by the sensor are continuous time series data. There are 6 sensors,  $ax$ ,  $ay$ ,  $az$ ,  $gx$ ,  $gy$ , and  $gz$ , respec-

tively. The acceleration data and gyroscope data represent three directions, respectively [23, 24].

In the traditional method based on machine learning, we need to analyze the original data and extract features manually. However, in the end-to-end deep learning method, there is no need to extract any manual features, and the data can be sent to the deep learning model only by preprocessing.

As can be seen from Figure 4, there are large differences in each data. For example, the value of  $ay$  is  $[0, 25000]$ , but the value of  $ax$  is  $[-25000, 25000]$ . These collected data have different dimensions and units. If the data is analyzed directly without processing, it will affect the results of data analysis. In order to eliminate the dimensional impact between indicators, data standardization processing is required. Standardization processing can solve the data magnitude gap between data indicators due to measurement units and other reasons. After data standardization, the values of each feature of the original data are in the same order of magnitude, so that the comprehensive comparison and evaluation of each feature data can be carried out, and the Min-Max Normalization is used to transform the original data into  $[0,1]$ . The process can be expressed as follows:

$$(\theta_i, R_i) = \frac{(\theta_i^D, R_i^D) - \min(\theta_i^D, R_i^D)}{\max(\theta_i^D, R_i^D) - \min(\theta_i^D, R_i^D)}, \quad (1)$$

where  $(\theta_i^D, R_i^D)$  is a sequence,  $\min(\theta_i^D, R_i^D)$  is the minimum value in  $(\theta_i^D, R_i^D)$ , and  $\max(\theta_i^D, R_i^D)$  is the maximum value in  $(\theta_i^D, R_i^D)$ . In the actual detection, the time delay of normalization is much less than the real detection delay, which can be ignored.

After Min-Max Normalization, we get time series data in the range of  $[0, 1]$  and the data shape is  $[T, \text{features}]$ ,  $T$  represents the time length, and features represents the number of features (MPU 6050 collects 6 features). However, deep learning is used to deal with the task of supervised learning, time series data cannot be directly used for supervised learning, so we need to transform the time series data.

Firstly, the sliding window is used to segment the timing data. In the experiment, the device collects data every 20 ms. Figure 5 shows the segmentation process: the total time length is  $T$  and the window size is  $t$ . After segmentation, multiple time slices are obtained, and the shape of each slice is  $[t, \text{features}]$ . Features are the feature number collected by the device. Here, it is 6.

After sliding window processing, the original time series data is divided into  $n$  slice data, which will be used as feature data. In order to reduce the time delay of real-time monitoring, it is far from enough to classify the actions at the current time. Our task is to infer the possible actions of the user at the next time, so we take the state of time  $t$  as the label of time  $t-1$ .

As shown in Figure 6, the original data shape is  $[\text{sample}, 7]$ , sample is the length of the sliding window and the data contains 6 features and 1 label. In Figure 5, we have demonstrated the segmentation process. So, we assign each slice data a label for the next time. Through these steps, we transform raw data

|   |   |
|---|---|
| AX: <input type="text" value="-740"/>           | AY: <input type="text" value="-840"/>         |
| AZ: <input type="text" value="-17928"/>         | GX: <input type="text" value="783"/>          |
| GY: <input type="text" value="-1137"/>          | GZ: <input type="text" value="8726"/>         |
| Port: <input type="text" value="COM1"/>         | Baud rate: <input type="text" value="15200"/> |
| <input type="button" value="Open serial port"/> |   |

FIGURE 2: User interface.

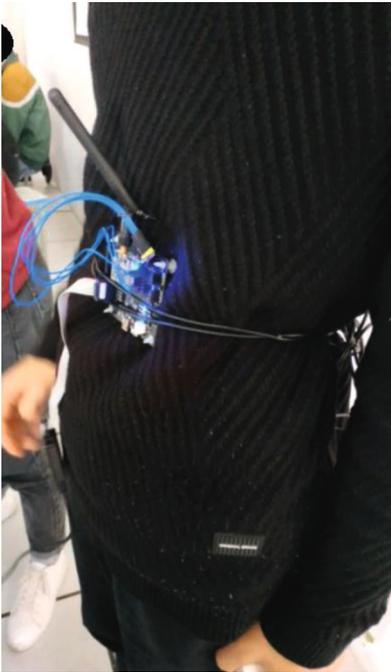


FIGURE 3: Diagram of wearing.

TABLE 1: Physiological statistics.

| Parameters  | Mean | Min  | Max  |
|-------------|------|------|------|
| Age (year)  | 20   | 19   | 21   |
| Weight (kg) | 65   | 45   | 75   |
| Height (m)  | 1.7  | 1.56 | 1.78 |

into feature  $TrainX$  and label  $TrainY$ . After segmentation, each data in  $TrainX$  has overlapping parts. But each data is independent, and they will not affect each other. This process is presented in Figure 6.

When we get  $TrainY$ , we need to encode  $TrainY$ . In this experiment, there are only four movements: walking, sitting down, running, and climbing stairs. We use one-hot encoding, as shown in Figure 7.

## 5. Time Series Convolution Network Model Based on Self-Attention

The current research on human activity recognition is to establish neural network model to detect the user's current activation, but they do not infer the user's activation at the next moment. Therefore, time series is added to the model to predict the user's activation at the next moment by learning from the user's previous sensor data.

In this paper, we establish a time series convolution network model based on self-attention (ATCRNN). The model first extracts the features between 6-axis sensor data through CNN, then sends the extracted features to LSTM and self-attention, respectively, for long-time and short-time feature extraction, and finally fuses the features and outputs them through the full connection layer.

**5.1. Feature Extract Based on CNN.** CNN temporal convolution layer is designed to extract the features from multidimensional data. This layer comprises convolution layer, pooling layer, and FC layer. The shape of the input data is  $[n, v]$ , where  $n$  represents the length of the time series and  $v$  represents the number of features.

Figure 8 shows the complete structure of the CNN layer. A whole CNN layer consists of convolution operation and Max-pooling operation. Convolution operation is the core of CNN layer. Different types of features can be extracted by convolution kernel, and each layer of data is usually convoluted by multiple convolution kernels [25, 26]. In this paper, two-dimensional convolution is used to extract multi-dimensional features according to the following formula:

$$X_c^{l+1} = f\left(W_c^{l+1} * X_c^l + b_c^{l+1}\right), \quad (2)$$

where  $X_c^l$  is the input data,  $X_c^{l+1}$  is the data with convolution,  $*$  denotes the convolution, and  $f$  is an activation function. Here, we use ReLU as the activation function:

$$\text{ReLU} : \sigma(x) = \max(0, x). \quad (3)$$

In order to reduce the number of features and keep the local invariance, the pool layer is followed by the convolution operation. The calculation formula of pooling layer is as follows:

$$a_j^l = \sigma\left(b_j^l + \beta_j^l f_{\text{down}}\left(a_j^{l-1}\right)\right). \quad (4)$$

In (4),  $\beta$  is the pooling weight, and  $f_{\text{down}}()$  is the pooling function. The common pooling functions are the maximum pooling function and the average pooling function, where  $k$  is the size of the pooling core.

$$f_{\text{down}}(\alpha) = \max(\alpha_{i,j}), \quad i, j \in [0, k], \quad (5)$$

$$f_{\text{down}}(\alpha) = \frac{1}{k \times k} \left( \sum_i \sum_j \alpha_{i,j} \right), \quad i, j \in [0, k]. \quad (6)$$

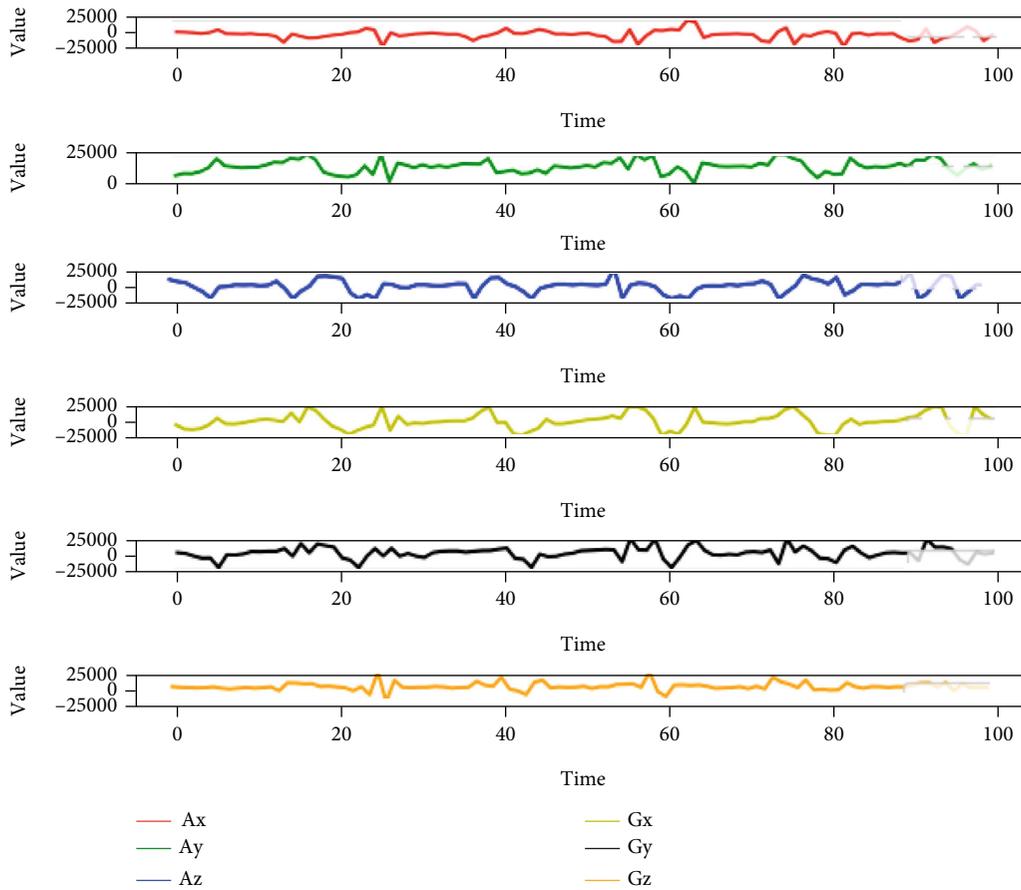


FIGURE 4: Sensing data.

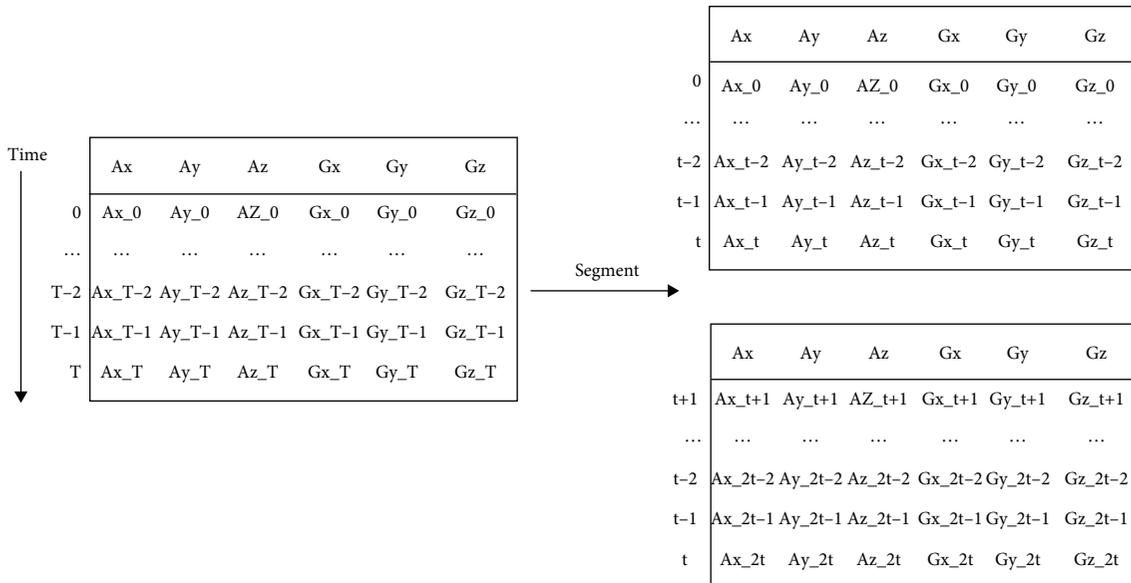


FIGURE 5: Segmentation process.

In training, the back-propagation gradient descent algorithm is used to adjust the parameters. CNN’s unique local receptive field and weight sharing mechanism limit the number of parameters to be trained within a certain range, which improves the training performance [27, 28].

5.2. Long-Time Extract Based on LSTM. The feature extracting of sensor data processed by CNN layer is input into LSTM layer to learn the long time series features. In the LSTM clock, the activation value of each neural unit at the previous time point is multiplied by the weight coefficient

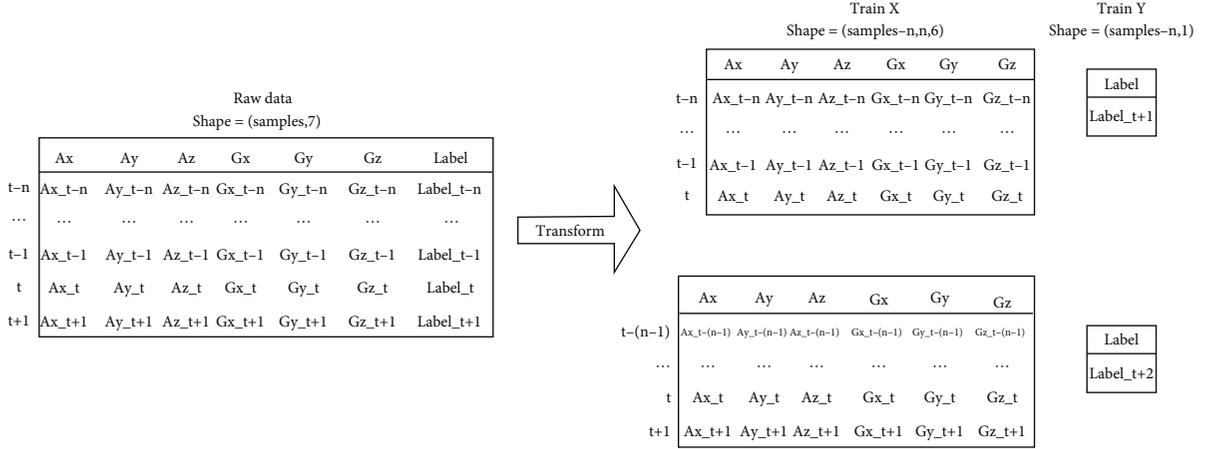


FIGURE 6: Data segmentation.

| Motion          | Value | One-hot code |
|-----------------|-------|--------------|
| Walk            | 0     | 0000         |
| Sit down        | 1     | 0001         |
| Run             | 2     | 0100         |
| Take the stairs | 3     | 1000         |

FIGURE 7: Encoding.

and then added back to the current time point, which exerts the influence of past activation function on it, which is equivalent to having memory. Each hidden layer of LSTM is recursively connected by a series of memory blocks, one memory block corresponds to a memory unit, and each memory unit contains three gates: input gate, output gate, and forgetting gate, which can read, write, and reset the memory unit, respectively, so as to flexibly control the information transmission between different memory units. The state of LSTM memory unit is updated by the following formula [29, 30]:

$$i_t = \sigma(W^i X_t + V^i h_{t-1} + b^i), \quad (7)$$

$$f_t = \sigma(W^f X_t + V^f h_{t-1} + b^f), \quad (8)$$

$$o_t = \sigma(W^o X_t + V^o h_{t-1} + b^o), \quad (9)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W^c X_t + V^c h_{t-1} + b^c), \quad (10)$$

$$h_t = o_t \otimes \tanh(c_t). \quad (11)$$

Among them,  $i_t$ ,  $f_t$ ,  $o_t$ ,  $c_t$ , and  $h_t$  represent the outputs of input gate, forgetting gate, output gate, control unit, and memory unit at time  $t$ , respectively.  $b^i$ ,  $b^f$ ,  $b^o$ , and  $b^c$  are the corresponding offset vectors.  $W^i$ ,  $W^f$ ,  $W^o$ ,  $W^c$ ,  $V^i$ ,  $V^f$ ,  $V^o$ , and  $V^c$  are weight matrixes.

The structure of LSTM network is designed as two hidden layers. Each hidden layer corresponds to an LSTM cell, which is expressed as *LSTM-cell1* and *LSTM-cell2* in turn.

When training the model, in order to achieve the trade-off between training efficiency and effect, the sensor data are input in batches. As a super parameter, the batch size can be selected by multiple assignment and test effect. A training iteration includes forward and back-propagation processes, and the parameters are updated once every iteration until the accuracy and loss tend to converge.

**5.3. Short-Time Extract Based on Self-Attention.** The feature mapping of inertial sensor data processed by LSTM network layer is input into the attention layer. The main body of the attention layer is the attention unit, but before the data enters the attention unit, it needs to be processed by the fully connected unit. Given that the input data size is  $(n, t, f)$ , the function of the fully connected unit is to map  $F$  features to  $F * \text{features}$ , which are the linear combination of the original  $F$  features [31–33]. This process can be used as a feature and processing to improve the resolution of attention coding in the next step. The idea of attention coding unit is to encode each time point based on the characteristics of each time point and obtain the higher level expression of features. The coding principle is as follows:

$$e_{ij} = a(S_{t-1}, h_{ij}), \quad (12)$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^F \exp(e_{tk})}, \quad (13)$$

$$C_t = \sum_{j=1}^F \alpha_{tj} h_j. \quad (14)$$

In (12), (13), and (14),  $S_t$  is the encoder state at time  $t$ ,  $h_{ij}$  represents the  $j$ th feature at time  $t$ ,  $a$  is the conversion function,  $e_{tj}$  is the influence factor of the  $j$ th feature at time  $t$ ,  $\alpha_{tj}$  is the coding weight, and  $C_t$  is the coding result at time  $t$ .

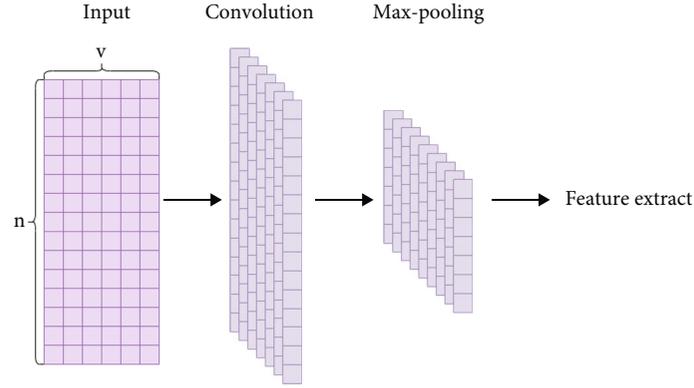


FIGURE 8: CNN layer construction.

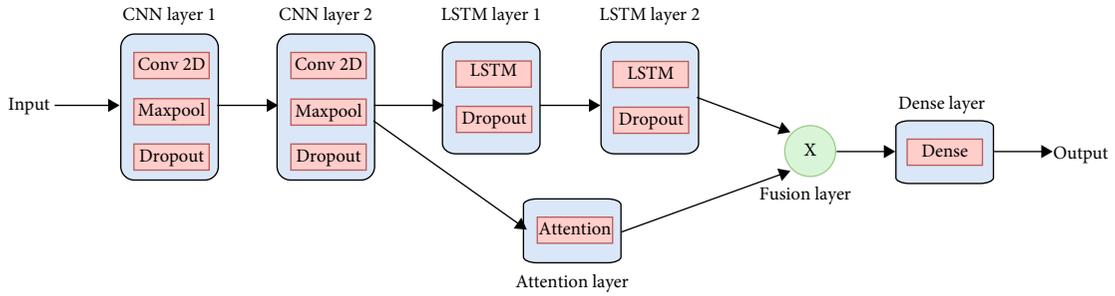


FIGURE 9: Network model structure.

5.4. *Feature Fusion.* After the extraction of long-time and short-time features, we fuse the features at fusion layer as follows:

$$\widetilde{X}_t = \tanh(F_{\text{lstm}}, F_{\text{attention}}), \quad (15)$$

where  $F_{\text{lstm}}, F_{\text{attention}}$  is the features extracted from LSTM layer and self-attention layer. Then, we use the full connection layer to infer the label of the activity. The activation function of the output is

$$\tanh(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (16)$$

Since we use one-hot to encode the label, so categorical\_crossentropy is used to calculate the loss.

$$\text{Loss}(\theta) = -\sum_{i=1}^m y_i \cdot \log \hat{y}_i. \quad (17)$$

Figure 9 shows the complete structure of ATCRNN network. Attention time series convolution network model (ATCRNN) utilizes CNN, LSTM, and attention mechanism. Therefore, we designed the following network structure: the time series data collected by the sensor is sent to the 2-D CNN layer for feature extraction, and the time series features of different lengths are extracted by LSTM layer and self-attention layer. Then, features extracted by LSTM and self-

TABLE 2: Experiment parameters.

| Parameters    | Value |
|---------------|-------|
| Batch_size    | 128   |
| Epochs        | 100   |
| CNN unit      | 100   |
| LSTM unit     | 50    |
| Learning_rate | 0.001 |
| Time_step     | 50    |

attention will be fused in fusion layer. Finally, the model connect to dense layer to get the output.

## 6. Experimental Analysis

6.1. *Parameters and Test Environment.* This experiment runs on the platform of Ubuntu 16.04 and is trained by Keras framework through CPU. In the process of training, the corresponding super parameters are obtained through several parameter tests and screening. The parameters are set in Table 2 as follows:

Firstly, the influence of the number of sensors and the wearing position on the accuracy of human activity recognition is considered. We used a sensor that was worn on the experimenter's waist. The activities are divided into four categories: walking, sitting, running, and climbing stairs. No feature engineering is carried out on the collected data. The data is divided according to the time sequence. The last 10% of the training dataset is taken as the test dataset.

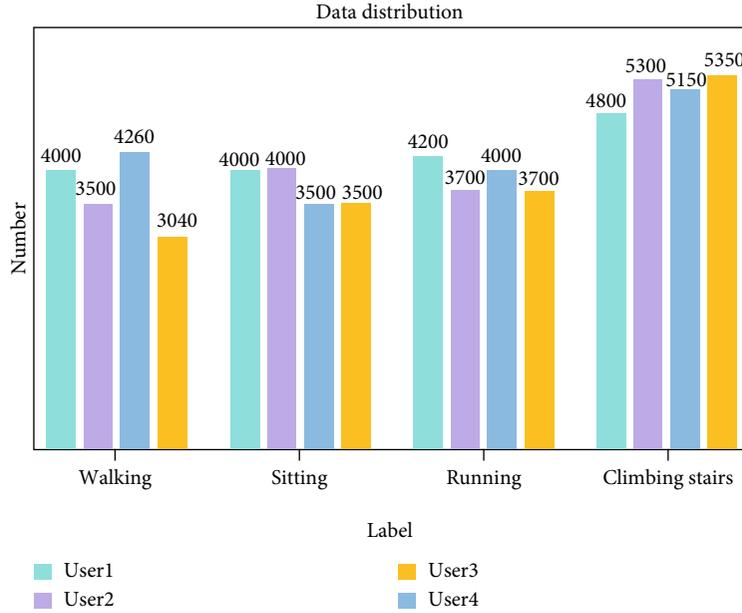


FIGURE 10: Data distribution.

6.2. *Dataset.* The dataset was obtained by four volunteers. The dataset contains 6550 pieces of data, including four types of actions: walking, sitting, running, and climbing stairs. The distribution of various data is shown in Figure 10. The dataset can be considered as a balanced dataset.

6.3. *Model Construction.* ATCRNN is built by Keras and the detail structure are shown in Figure 11. Intel Core i7 dual core processor is used in the test, which is carried out under MacOS 64 bit operating system. The model runs in Anaconda environment of Python 3.6 kernel.

6.4. *Evaluation Metrics.* In order to evaluate the performance of the algorithm, this paper uses accuracy, precision, recall, and F1 score to evaluate the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (18)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (19)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (20)$$

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (21)$$

Among them, FN (false negative) is positive sample judged as negative sample, FP (false positive) is negative sample judged as positive sample, TN (true negative) is negative sample judged as negative sample, and TP (true positive) is positive sample judged as positive sample.

6.5. *Analysis of Model Results.* Due the data, we got is a balanced dataset, so we use accuracy as the main evaluation metric. In the training set and test set, the accuracy is 96.24%

and 97.15%, respectively. The confusion matrix on the test set is presented in Table 3 as follows:

We also compare the ATCRNN algorithm with other models. They are stacked LSTM model, stacked CNN model, and CNNLSTM model. The network structure of ATCRNN model is composed of CNN network, LSTM network, and attention network; the STACKLSTM model is composed of two LSTM networks; the StackCNN model is composed of two CNN networks; the CNNLSTM model is composed of CNN network and LSTM network. The four models involved in the comparison were trained with the same data, and the parameters were adjusted. The results in Table 4 are the best results of each model.

Through comparative evaluation, the accuracy of ATCRNN model is higher than those of the other three models. In the prediction of walking posture, both the ATCRNN model and StackCNN model have reached 100% prediction accuracy. The prediction accuracy of ATCRNN model for running is high. The prediction accuracy of sitting posture is 0.0067 lower than that of the optimal model. The accuracy of stair climbing posture is 0.0225 lower than the optimal result. But overall, the accuracy rate is 97.15%. It can also be seen from the results. In addition, attention can better extract time features and improve the accuracy of prediction.

6.6. *Loss Compare.* Figure 12 shows the loss comparison of the four algorithms in each epoch. It can be seen from the figure that the loss of CNNLSTM model and ATCRNN model is minimum and more stable at the beginning. The loss of StackLSTM is the most unstable. In general, the loss of ATCRNN and CNNLSTM is smaller and more stable, and the loss of ATCRNN is 0.03 less than that of CNNLSTM. The convergence of ATCRNN model is the best while that of CNNLSTM is the worst.

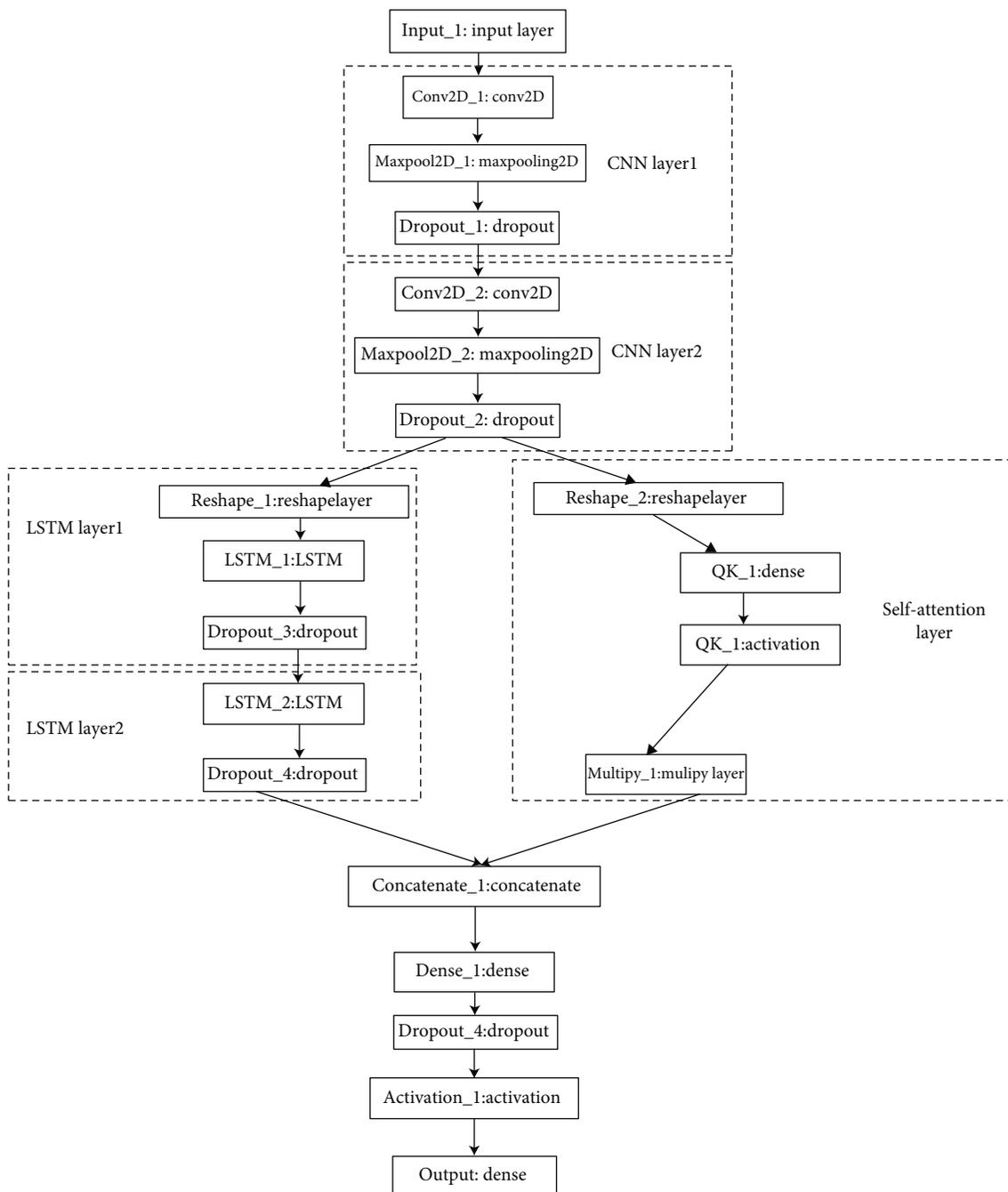


FIGURE 11: The detail structure of ATCRNN.

TABLE 3: Confusion matrix.

|        |                 | Predict |         |         |                 | Accuracy |
|--------|-----------------|---------|---------|---------|-----------------|----------|
|        |                 | Walking | Sitting | Running | Climbing stairs |          |
| Actual | Walking         | 230     | 0       | 0       | 0               | 1.0      |
|        | Sitting         | 10      | 290     | 0       | 0               | 0.9666   |
|        | Running         | 0       | 10      | 290     | 0               | 0.9666   |
|        | Climbing stairs | 0       | 0       | 15      | 385             | 0.9625   |
|        | Recall          | 0.9583  | 0.9666  | 0.9508  | 1.0000          |          |

TABLE 4: Model comparison result.

| Algorithm | Walking | Sitting | Running | Climbing stairs | Accuracy |
|-----------|---------|---------|---------|-----------------|----------|
| ATCRNN    | 1.0000  | 0.9666  | 0.9666  | 0.9625          | 0.9715   |
| StackLSTM | 0.9391  | 0.9966  | 0.9600  | 0.9100          | 0.9487   |
| StackCNN  | 1.0000  | 0.8366  | 0.9600  | 0.9850          | 0.9437   |
| CNNLSTM   | 0.9782  | 0.9733  | 0.9600  | 0.9625          | 0.9674   |

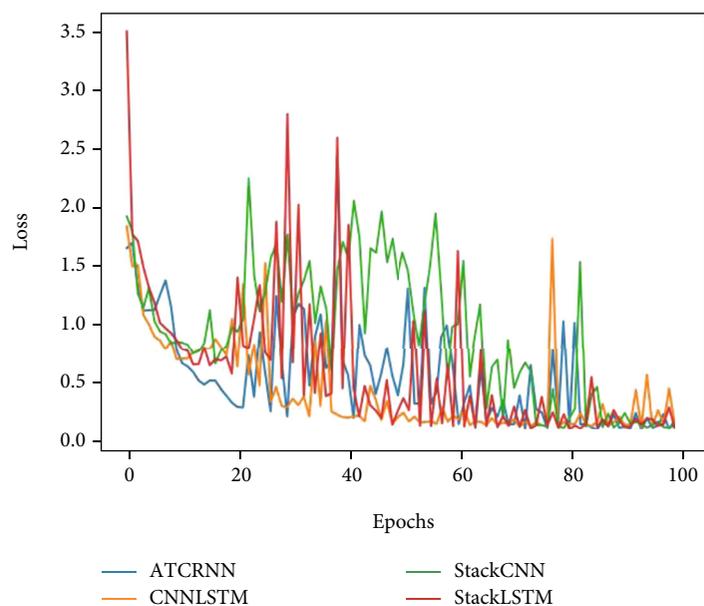


FIGURE 12: Loss compare.

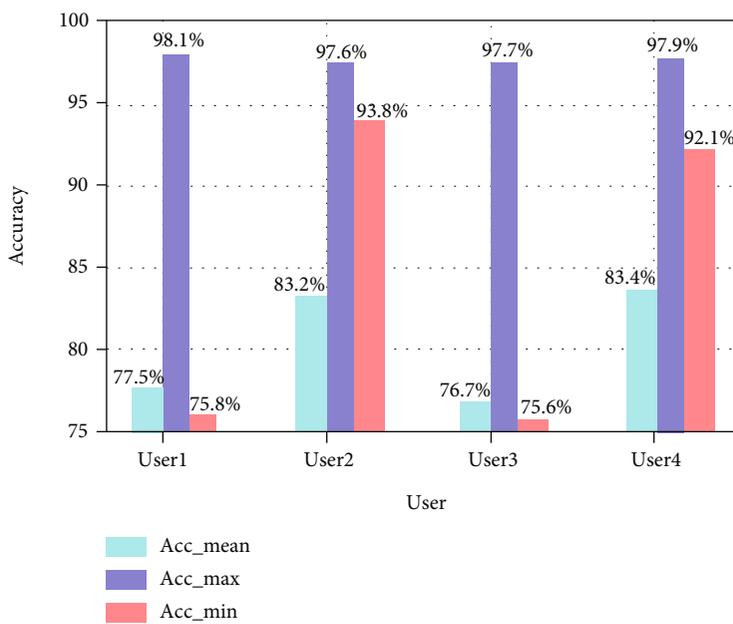


FIGURE 13: The influence of different experimenters.

**6.7. User Impact.** Considering the influence of individual users on the recognition effect, we conducted experiments on four volunteers. Based on the existing recognition model, we tested the recognition accuracy of four volunteers, and the results are shown in Figure 13. The recognition accuracy of volunteers' gestures is slightly different due to the influence of user's body shape and position. Among the four volunteers, the lowest recognition accuracy was 97.6%, the highest was 98.1%, and the average accuracy was 97.82%, which verified that the recognition model also has a good recognition ability for different user gestures.

## 7. Conclusion

This paper designs a real-time monitoring method based on wearable devices. This method uses wireless body area network for health care. Firstly, this paper introduces the overall framework of wearable device, introduces the structure and acquisition process of the device in detail, and gives a specific data preprocessing method based on a sliding window. Secondly, it will not meet the requirements of real-time monitoring if only classifying the action at the current time. So, this paper designs an end-to-end neural network model called ATCRNN to infer the actions to be made by users at the next time according to the data of the past few times. This model uses CNN and RNN to extract the spatial and temporal features of data and captures the context characteristics through self-attention. Finally, volunteers wore equipment to participate in the experiment. The activity categories in the experiment are walking, sitting down, running, and climbing stairs. The accuracy of behavior inference reached 97% with ATCRNN.

In the future, we will try to add some more complex data, so that our wearable devices can infer more movements in real time, and finally help patients recover.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the Natural Science Foundation of China (Grant 61871412), in part by the Wuhu City Science and Technology Plan Project (2021cgl7), and in part by the Key Research and Development Projects in Anhui Province (Grant 202004a05020002).

## References

- [1] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402–427, 2013.
- [2] C. Zhao, H. Zhang, F. Chen, S. Chen, C. Wu, and T. Wang, "Spatiotemporal charging scheduling in wireless rechargeable sensor networks," *Computer Communications*, vol. 152, pp. 155–170, 2020.
- [3] A. Argyriou, A. C. Breva, and M. Aoun, "Optimizing data forwarding from body area networks in the presence of body shadowing with dual wireless technology nodes," *IEEE Transactions on Mobile Computing*, vol. 14, no. 3, pp. 632–645, 2015.
- [4] Y. Zhou, Y. Y. Liu, C. M. Wang, J. Chen, Q. P. Chen, and Y. J. Wei, "Contrast research of several human motion detection algorithm," *Journal of Jilin University*, vol. 6, 2009.
- [5] A. Jain and V. Kanhangad, "Human activity classification in smartphones using accelerometer and gyroscope sensors," *IEEE Sensors Journal*, vol. 18, no. 3, pp. 1169–1177, 2018.
- [6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," *Esann*, vol. 3, 2013.
- [7] Z. Feng, L. Mo, and M. Li, "A random forest-based ensemble method for activity recognition," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5074–5077, IEEE, Milan, Italy, 2015.
- [8] M. Guo, Z. Wang, N. Yang, Z. Li, and T. An, "A multisensor multiclassifier hierarchical fusion model based on entropy weight for human activity recognition using wearable inertial sensors," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 1, pp. 105–111, 2019.
- [9] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.
- [10] C. L. Yang, Z. X. Chen, and C. Y. Yang, "Sensor classification using convolutional neural network by encoding multivariate time series as two-dimensional colored images," *Sensors*, vol. 20, no. 1, p. 168, 2020.
- [11] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.
- [12] C. A. Ronao and S. B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.
- [13] H. Hikawa and K. Kaida, "Novel FPGA implementation of hand sign recognition system with SOM-Hebb classifier," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 153–166, 2015.
- [14] W. Tang and E. S. Sazonov, "Highly accurate recognition of human postures and activities through classification with rejection," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 309–315, 2014.
- [15] N. Carbonaro, G. D. Mura, F. Lorussi, R. Paradiso, D. de Rossi, and A. Tognetti, "Exploiting wearable goniometer technology for motion sensing gloves," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1788–1795, 2014.
- [16] G. Orengo, A. Lagati, and G. Saggio, "Modeling wearable bend sensor behavior for human motion capture," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2307–2316, 2014.
- [17] L. Wang, T. Gu, X. Tao, and J. Lu, "A hierarchical approach to real-time activity recognition in body sensor networks," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 115–130, 2012.
- [18] Z. He and X. Bai, "A wearable wireless body area network for human activity recognition," in *2014 Sixth International*

- Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 115–119, IEEE, Shanghai, China, 2014.
- [19] V. Bijalwan, V. B. Semwal, G. Singh, and R. G. Crespo, “Heterogeneous computing model for post-injury walking pattern restoration and postural stability rehabilitation exercise recognition,” *Expert Systems*, no. article e12706, 2021.
- [20] W. Jiang and Z. Yin, “Human activity recognition using wearable sensors by deep convolutional neural networks,” in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1307–1310, New York, 2015.
- [21] S. M. Lee, S. M. Yoon, and H. Cho, “Human activity recognition from accelerometer data using convolutional neural network 2017,” in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 131–134, IEEE, Jeju, Korea (South), 2017.
- [22] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “LSTM: a search space odyssey,” *IEEE Transactions On Neural Networks And Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.
- [23] Y. Yu, D. Wang, R. Zhao, and Q. Zhang, “RFID based real-time recognition of ongoing gesture with adversarial learning,” in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pp. 298–310, New York, 2019.
- [24] F. Xiao, L. Pei, L. Chu et al., “A deep learning method for complex human activity recognition using virtual wearable sensors,” in *Spatial Data and Intelligence*, pp. 261–270, Springer, Cham, 2020.
- [25] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015.
- [26] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, “TSE-CNN: a two-stage end-to-end CNN for human activity recognition,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 1, pp. 292–299, 2020.
- [27] F. Chen, Y. Tang, C. Wang et al., “Medical Cyber-Physical Systems: A Solution to Smart Health and the State of the Art,” *IEEE Transactions on Computational Social Systems*, pp. 1–8, 2021.
- [28] S. M. Lee, S. M. Yoon, and H. Cho, “Human activity recognition from accelerometer data using convolutional neural network,” in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 131–134, IEEE, Jeju, Korea (South), 2017.
- [29] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang, “A Siamese long short-term memory architecture for human re-identification,” in *Computer Vision – ECCV 2016*, pp. 135–153, Springer, Cham, 2016.
- [30] D. Deb, A. Ross, A. K. Jain, K. Prakah-Asante, and K. V. Prasad, “Actions speak louder than (pass) words: passive authentication of smartphone users via deep temporal features,” in *2019 international conference on biometrics (ICB)*, pp. 1–8, IEEE, Crete, Greece, 2019.
- [31] D. Andrei and M. Hasler, “Investor attention and stock market volatility,” *The Review of Financial Studies*, vol. 28, no. 1, pp. 33–72, 2015.
- [32] M. Ji, W. Joo, K. Song, Y. Y. Kim, and I. C. Moon, “Sequential recommendation with relation-aware kernelized self-attention,” *Proceedings of the AAAI Conference On Artificial Intelligence*, vol. 34, no. 4, pp. 4304–4311, 2020.
- [33] Z. Cao, R. Wang, X. Wang, Z. Liu, and X. Zhu, “Improving human pose estimation with self-attention generative adversarial networks,” in *2019 IEEE international conference on Multimedia & Expo Workshops (ICMEW)*, pp. 567–572, IEEE, Shanghai, China, 2019.