EasyChair Preprint

№ 7218

Fine-Grained Activity Recognition Based on Features of Action Subsegments and Incremental Broad Learning

Shi Chen, Sheng Wu, Licai Zhu and Hao Yang

December 16, 2021

Fine-grained Activity Recognition Based on Features of Action Subsegments and Incremental Broad Learning

Shi Chen[1,2][0000-0003-1443-1760] Sheng Wu[1,2][0000-0002-4391-3892]

Licai Zhu[2,1][0000-0003-1702-7528] Hao Yang[2,1,3,4][0000-0001-6521-3080]

[1] Nanjing Tech University, Nanjing, Jiangsu, 211816, China
[2] Yancheng Teachers University, Yancheng, Jiangsu, 224001, China
[3] Jiangsu Provincial Key Constructive Laboratory for Big Data of Psychology and Cognitive Science, Yancheng Teachers University, Yancheng, Jiangsu, 224002, China
[4] Suzhou Research Institute, University of Science and Technology of China, Suzhou, Jiangsu, 215123, China
Corresponding author: Licai Zhu, hobbyc@163.com

Abstract. Human activity recognition using MEMS on mobile devices has become one of the most compelling solutions owing to the miniaturization of sensors. A crucial challenge is to recognize precisely activities when they are changing. Sliding window is a type of common methods. However, the interference of historical data in the sliding window is harmful to insight into changing of actions or uncommon behaviors. This paper proposes a fine-grained activity recognition method and designs a corresponding system *farer*. It employs features of action subsegments and incremental broad learning to precisely distinguish the alterations of actions and abnormal movements. Firstly, *far*er achieves the accurate segmentation of activities as data preprocessing. A neighborhood extreme value method (NEV) is adopted to avoid the intervention of peaks and valleys of data. Secondly, the current action is partitioned to fine-grained subsegments to elaborately abstract subtle features. We propose a feature extraction technique based on adjacent difference (FETAD), and furthermore reduce its resulting dimension through the complete two-dimensional principal component analysis (C2DPCA). Finally, broad learning theory is employed to construct the activity recognition model, especially incremental learning for unusual behaviors. Extensive experiments demonstrate that *farer* could accurately recognize activities when they abruptly change, and its performance is considerable stability. Meanwhile, it can quickly establish a valid incremental model that only needs a short sampling time for special activities. The overall accuracy of *farer* is 97.91% with 90.14% for changed activities, which is far superior to the current mainstream methods.

Keywords: Fine-grained recognition; Broad learning; Action subsegments; Incremental model

1 Introduction

Human Activity Recognition (HAR) has been widely applied in some scenarios, e.g. industry and medicine [1]. For instance, it could be used for early warning when factory operators carry out dangerous behaviors, and exhibiting the accuracy degree of mobility impairments patients' performed actions for physical therapy. The basis pattern of HAR

is to collect actual data of specific activities and then achieve accurate recognition through comparison and classification. In generally, feature extraction by sliding window is an effective activity recognition paradigm. The basic essence of this type of method is to design a sliding window which contains the perception data of the activity to be recognized, then extract features of the window data to represent the activity, and design an effective classifier for activity recognition. However, in the above special scenarios, most of them require the deployment of dedicated sensing systems, such as IoT devices and sensor networks [2-4], resulting in high cost and poor universality.

The development of wireless communication technology [5] and the miniaturization of sensors [6] have enabled smart devices [7] embedded with many MEMS sensors (such as smart phones, tablets, etc.) to have good data collection capabilities. In the meanwhile, in-depth research in the field of deep learning has further improved the accuracy of activity recognition [8]. However, the popularization of mobile devices puts forward higher requirements for activity recognition in complex scenarios. Particularly, mobile devices must quickly and accurately perceive activity changes, and recognition methods must be oriented to the special behavior habits of different users. To this date, the main challenges that activity recognition still faces are as follows: 1) The raw data collected by the accelerometer contains much noise. 2) The influence of historical data on current activity data. Too much historical data is not conducive to the recognition of activity changes. 3) Special behavior habits are not easy to accurately recognize.

To address the above challenges, in this paper, we propose a fine-grained activity recognition method and designs a corresponding system *farer*. The system implements effective preprocessing of the source data by smoothing and filtering the sampled data and segmenting activities into individual actions. Then deeply mine the features of actions through fine-grained subsegment division, feature extraction and dimensionality reduction. Finally, an incremental activity recognition model based on broad learning (BL) is constructed to realize accurate activity recognition and satisfy incremental model update. The main contributions of this paper are summarized as follows:

1) Realize accurate segmentation of activities. By smoothing and filtering the source data, a neighborhood extreme value method is proposed to avoid the interference of peaks and valleys.
2) Deeply mine the features of action subsegments. The action data is partitioned to fine-grained subsegments according to changes of acceleration, and a feature extraction technique oriented to adjacent difference is designed.
3) The *farer* system based on BL is designed with stable performance and good practicability. After a series of experimental verifications, the system has a high recognition rate for different activities under the condition of stable activities, with an overall recognition rate of 97.91%. Under the condition of changed activities, the recognition accuracy is 90.14%, far exceeding other methods.

2 Related Work

Recognizing human activities based on sensor data is essentially a pattern classification problem. When using a sliding window for activity recognition, it is required to process the noise generated during the sampling process, select an appropriate sliding window, and design an effective activity classifier. These are all key factors that determine the

recognition performance. This section mainly introduces the current research status of three aspects in the case of activity recognition: the processing of data noise, the setting of sliding window and the construction of classifier model.

Since the data collected by the sensor often contains high-frequency noise, it needs to be processed, otherwise it will affect the accuracy of activity recognition. Yang et al. [9] used Gaussian filtering algorithm to eliminate the influence of noise. Garcia-Ceja et al. [10] used the average smoothing method, replacing each original data with the average of two adjacent data points. Khan et al. [11] used a third-order moving average filtering algorithm to remove noise. These methods cannot eliminate the interference extreme points to a great extent.

The collection of human activity data often takes a long time. In the activity recognition stage, the sensor data stream needs to be segmented by windows. Because the fixed-size sliding window segmentation technology is simple to operate, it is adopted by most researches. In the terms of sliding window setting, Fida et al. [12] tested the window of 0.5 seconds to 3 seconds, and achieved the best result in 1.5 seconds. Elsts [13] adopted a 2.56 second sliding window to design the energy-saving activity recognition framework. Shuvo [14] and Xia [15] adopted a sliding window of 2.56 seconds with a step length of 1.28 seconds, achieving the recognition accuracy of more than 95%. Cha et al. [16] adopted a window length of 1 to 4 seconds and found that using 4 seconds achieved the best accuracy of 96.1%. Pienaar et al. [17] adopted a large window of 10 seconds with a step length of 1 second to segment data and achieved the recognition accuracy of 94%. The above methods verify that the selection of the action window will greatly affect the recognition performance of the activity. The traditional method of fixed-size sliding window is oriented to the recognition of a single stable activity, while ignoring the change of the activity.

In the terms of classifier construction, researchers manually extract features from activity sensor data and employ them in various traditional machine learning algorithms. Since these data fragments cannot adequately represent complex human activities, they have become the performance bottleneck of the classifier [18]. To further improve the accuracy of activity recognition, methods based on deep learning are employed by more and more people, such as Convolutional Neural Network (CNN) [19], Long Short-Term Memory (LSTM) [20] and their joint improved models CNN-LSTM [21] and ConvLSTM [22]. Although these models show good performance, their designs are more complex. In addition, the amount of calculation is large and hardware requirements are high. More importantly, these models are constructed based on training data, so they have poor perception of activity changes and lack robustness [23].

To solve the problems of accurate recognition and flexibility of activities, we propose effective data preprocessing and employ fine-grained features of action subsegments, which improve the accuracy of activity recognition, especially for changed activities. Incremental model construction based on broad learning is also proposed to realize the incremental update of the special activity behavior, without using the source data to retrain the entire model.

3 Proposed Method

For complex activity situations, we propose a fine-grained activity recognition method

based on features of action subsegments and incremental broad learning. The corresponding recognition system named *farer* is also designed. The system contains three sub-modules, namely data preprocessing, feature extraction based on fine-grained subsegment, and incremental recognition model based on broad learning. First, smooth and filter the sampled data, and design a peak and valley recognition method to accurately segment activities into individual actions. Then deeply mine the features of action subsegments through the fine-grained segmentation of the action data, targeted feature extraction and dimensionality reduction. Finally, build an incremental activity recognition model based on broad learning to realize the accurate recognition of activities and satisfy the incremental model update. The framework of *farer* is shown in Figure 1.

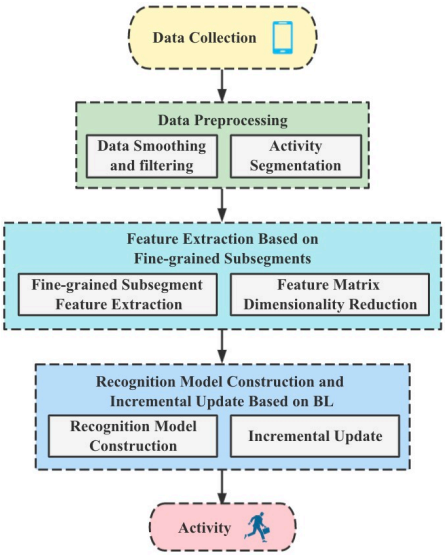

Fig. 1. *farer* framework.

3.1 Data Preprocessing

Data Smoothing and Filtering. According to the travel characteristics of pedestrians, the sensor data changes smoothly in a relatively short period. Since people cannot maintain a fixed posture when traveling, the sensor perceives irregular changes during the collection process, resulting in abnormal points. Without changing the trend of data changes, we employ the neighborhood smoothing method to filter the source data.

Assuming that the sampled data at time t is x_t, its neighborhood interval is $[t-\varphi, t+\varphi]$ and the interval range is $\mu = 2\varphi + 1$. Construct a k-order polynomial to fit the points in the interval. Denote s_0、s_1、$\cdots$、s_k as the polynomial coefficients, then the data x_t can be expressed as

$$x_t = s_0 + s_1 t + s_2 t^2 + \cdots + s_k t^k \tag{1}$$

The least square fitting of the neighborhood interval $X = \left(x_{t-\varphi}, \cdots, x_t, \cdots, x_{t+\varphi}\right)^T$ is calculated as

$$\hat{X} = P(P^T P)^{-1} P^T X \tag{2}$$

where,

$$P=\begin{pmatrix}1 & t-\varphi & (t-\varphi)^2 & \cdots & (t-\varphi)^k\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & t-1 & (t-1)^2 & \cdots & (t-1)^k\\ 1 & t & t^2 & \cdots & t^k\\ 1 & t+1 & (t+1)^2 & \cdots & (t+1)^k\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & t+\varphi & (t+\varphi)^2 & \cdots & (t+\varphi)^k\end{pmatrix}.$$

The method of smoothing and filtering the source data saves the change information of the signal and eliminates outliers, which makes the data curve smoother. After filtering, the value of x_t is $\hat{X}(\varphi+1)$.

Activity Segmentation. The periodicity of activities makes the acceleration data in the vertical direction sampled by the sensor show regular changes in peaks and valleys, so dividing activities by identifying peaks and valleys is the basic way of activity segmentation. The activity data is filtered to eliminate abnormal signal points, making the entire data stream smoother. However, there are still multiple extreme interference points at the peak and valley of activities, which seriously affects the accurate segmentation of activities.

We design a neighboring extremum value method (NEV) to more accurately identify the real peaks and valleys and avoid the interference of extreme points. Denote $X=(x_1,x_2,\ldots,x_n)$ as the acceleration sampling data in the vertical direction. Then,

1. Calculate extreme points.

Obtain all maximum points $X_{max}=(x_{max}^1,x_{max}^2,\ldots)$ and minimum points $X_{min}=\left(x_{min}^1,x_{min}^2,\ldots\right)$.

2. Filter extreme interference points.

We employ the altitude, prominence and isolation of the peaks and valleys to filter the noise at the extreme points.

Assuming that the altitude threshold is Γ_A, the prominence threshold is Γ_P and the isolation threshold is Γ_I. Denote $t(x)$ as the time (data volume) scale of point x. The specific filtering process is as follows:

1) Altitude. Eliminate lower peaks or shallower valleys.

- When the peak maximum point set X_{max} satisfies $x_{max}^i<\Gamma_A$, the point x_{max}^i is eliminated, i.e. $X_{max}-\{x_{max}^i\}$. By traversing X_{max}, Y_{max} that meets the predetermined altitude is obtained.
- When the valley minimum point set X_{min} satisfies $x_{min}^j>-\Gamma_A$, the point x_{min}^j is eliminated, i.e. $X_{min}-\left\{x_{min}^j\right\}$. By traversing X_{min}, Y_{min} that meets the predetermined altitude is obtained.

2） Prominence. Eliminate peaks with less convexity or valleys with less concavity.

- In the peak maximum point set Y_{max}, if there is a minimum natural number a that satisfies $y_{max}^i<y_{max}^{i-a}$, then denote $y_{min}^{j_a}$ as the minimum valley in the data interval

$[t(y_{max}^{i-a}), t(y_{max}^{i})]$. Meanwhile, if there is a minimum natural number b that satisfies $y_{max}^{i} < y_{max}^{i+b}$, then denote $x_{min}^{j_b}$ as the minimum valley in the data interval $[t(y_{max}^{i}), t(y_{max}^{i+b})]$. If $min\left\{y_{max}^{i} - y_{min}^{j_a}, y_{max}^{i} - y_{min}^{j_b}\right\} < \Gamma_P$, then the maximum point y_{max}^{i} is eliminated, i.e. $Y_{max} - \{y_{max}^{i}\}$. By traversing Y_{max}, Z_{max} that meets the predetermined prominence is obtained.

- In the valley minimum point set Y_{min}, if there is a minimum natural number c that satisfies $y_{min}^{i} > y_{min}^{i-c}$, then denote $y_{max}^{j_c}$ as the maximum peak in the data interval $[t(y_{min}^{i-c}), t(y_{min}^{i})]$. Meanwhile, if there is a minimum natural number d that satisfies $y_{min}^{i} > y_{min}^{i+d}$, then denote $x_{max}^{j_d}$ as the maximum peak in the data interval $[t(y_{min}^{i}), t(y_{min}^{i+d})]$. If $min\{y_{max}^{j_c} - y_{min}^{i}, y_{max}^{j_d} - y_{min}^{i}\} < \Gamma_P$, then the minimum point y_{min}^{i} is eliminated, i.e. $Y_{min} - \{y_{min}^{i}\}$. By traversing Y_{min}, Z_{min} that meets the predetermined prominence is obtained.

3） Isolation. Eliminate the smaller point of two peaks that are close in horizontal distance or the larger point of two valleys that are close in horizontal distance.

- When there are adjacent points z_{max}^{i} and z_{max}^{i+1} in Z_{max}, and the horizontal distance between the two points satisfies $t(z_{max}^{i+1}) - t(z_{\max}^{i}) < \Gamma_I$, eliminate the smaller maximum point, i.e. $Z_{max} - min\{z_{max}^{i}, z_{max}^{i+1}\}$. By traversing Z_{max}, X'_{max} that meets the predetermined isolation is obtained.
- When there are adjacent points z_{min}^{i} and z_{min}^{i+1} in Z_{min}, and the horizontal distance between the two points satisfies $t(z_{min}^{i+1}) - t(z_{\min}^{i}) < \Gamma_I$, eliminate the bigger minimum point, i.e. $Z_{min} - max\{z_{min}^{i}, z_{min}^{i+1}\}$. By traversing Z_{min}, X'_{min} that meets the predetermined isolation is obtained.

3. Segment activities into individual actions.

When the vertical acceleration direction is upward and gradually increases from 0, it is defined as the starting point of the action. Then the peak and valley are reached. After reaching valley, when the vertical acceleration direction is downward and decreases to 0, it is defined as the ending point of the action. Thus, the activity data is segmented into individual action data.

Different from the traditional activity recognition, we set the size of the sliding window to the size of the complete action segment, so each window has a different size. We call the sliding window here the action window. The action window only contains the data of the current action, without historical data, and it slides a complete action window every time. Our design avoids the influence of historical data on current data.

3.2 Feature Extraction Based on Fine-grained Subsegments

Fine-grained Subsegment Feature Extraction. To extract fine-grained features of segmented actions, we design an activity recognition method based on fine-grained subsegments. We perform fine-grained subsegment division of actions to realize fine-grained cognition of actions, that is, the action window is evenly divided into several subsegments. The fine-grained division of the action window shows the change of the behavior state.

In order to fully mine the action characteristics to realize the fine-grained cognition

of the action data, we design a feature extraction technique based on adjacent difference (FETAD) for 3 axes acceleration. The change of measured data is relatively stable in a short period due to the continuity of the action. Therefore, when the action is fine-grained divided, the difference in adjacent subsegments changes most smoothly. According to this principle, the steps of FETAD are as follows:

1) The action window is evenly divided into k_f subsegments and the length is l_m. The three-axis data vectors are $G_{l_i}^x = \left[g_{(i,1)}^x, g_{(i,2)}^x, \cdots, g_{(i,l_i)}^x\right]$, $G_{l_i}^y = \left[g_{(i,1)}^y, g_{(i,2)}^y, \cdots, g_{(i,l_i)}^y\right]$, $G_{l_i}^z = \left[g_{(i,1)}^z, g_{(i,2)}^z, \cdots, g_{(i,l_i)}^z\right]$, where $1 \le i \le k_f$.
2) Adopt the difference between the data in adjacent subsegments as the data of the previous subsegment, i.e. $G_{l_i}^x = G_{l_{i+1}}^x - G_{l_i}^x$, $G_{l_i}^y = G_{l_{i+1}}^y - G_{l_i}^y$, $G_{l_i}^z = G_{l_{i+1}}^z - G_{l_i}^z$, where $1 \le i < k_f$. Each axis gets $k_f - 1$ difference vectors.
3) Extract features for each difference vector of each coordinate axis. Denote n_f as the number of features to be extracted. The feature vector of the difference vector is $D_{l_i} = \left[d_i(1), d_i(2), \cdots, d_i(n_f)\right]$.
4) Combine the features of $k_f - 1$ difference vectors on each coordinate axis into a new feature vector. The feature vectors of the three coordinate axes are expressed as $D^x = \left[D_{l_1}^x, D_{l_2}^x, \cdots, D_{l_{k_f-1}}^x\right]^T$, $D^y = \left[D_{l_1}^y, D_{l_2}^y, \cdots, D_{l_{k_f-1}}^y\right]^T$ and $D^z = \left[D_{l_1}^z, D_{l_2}^z, \cdots, D_{l_{k_f-1}}^z\right]^T$.
5) Finally, the feature vectors of the three coordinate axes are combined into a two-dimensional matrix as $D^{xyz} = [D^x, D^y, D^z]^T$. The size is $3 \times [(k_f - 1) \times n_f]$.

Feature Matrix Dimensionality Reduction. To improve the speed of activity recognition, we further extract the effective information of the three-axis features. First, perform feature extraction on the combined matrix D^{xyz} of the x, y and z three axes. Aiming at the obtained two-dimensional feature matrix, we adopt complete two-dimensional principal component analysis (C2DPCA). C2DPCA reduces the dimensionality of the matrix from two aspects: row projection and column projection. Then, flatten the principal component matrix of the three axes to obtain a one-dimensional vector to meet the input requirements of BLS.

Denote D as the feature matrix set of N actions. The feature matrix of the i-th action is $D_i \in R^{p \times q}$, $i \in [1, N]$. To realize the complete two-dimensional principal component analysis of D_i, it needs to be projected from two angels of row and column. The column divergence matrix and row divergence matrix of D are formulated as $G_P = \sum_{i=1}^{N} (D_i - \overline{D})(D_i - \overline{D})^T$ and $G_Q = \sum_{i=1}^{N} (D_i - \overline{D})^T (D_i - \overline{D})$, where $\overline{D} = \frac{1}{N} \sum_{i=1}^{N} D_i$ is the average value of the feature matrix set D.

By choosing proper eigenvectors of the matrices G_P and G_Q, the projection of D_i is as dispersed as possible. The numbers of eigenvalues of G_P and G_Q are calculated as n_P and n_Q. The eigenvalues of G_P and G_Q are sorted in descending order. The eigenvalue set of G_P are $\alpha = \left[\alpha_1, \alpha_2, \cdots, \alpha_{n_P}\right]$, and the corresponding eigenvector set is

$[\mu_1, \mu_2, \cdots, \mu_{n_P}]$. The eigenvalue set of G_Q are $\beta = [\beta_1, \beta_2, \cdots, \beta_{n_Q}]$, and the corresponding eigenvector set is $[\nu_1, \nu_2, \cdots, \nu_{n_Q}]$. Since G_P and G_Q are both non-negative definite matrices, their eigenvalues are not less than zero.

If the first n_1 eigenvalues of the eigenvalue sequence α of G_P satisfy $\sum_{i=1}^{n_1} \alpha_i \geq \partial_P \cdot \sum_{i=1}^{n_P} \alpha_i$, where ∂_P is the column hash threshold. Select the eigenvectors corresponding to n_1 eigenvalues to form the column projection of D_i, i.e. $P = [\mu_1, \mu_2, \cdots, \mu_{n_1}]^T$. The size of P is $n_1 \times p$. Similarly, if the first n_2 eigenvalues of the eigenvalue sequence β of G_Q satisfy $\sum_{j=1}^{n_2} \beta_j \geq \partial_Q \cdot \sum_{j=1}^{n_Q} \beta_j$, where ∂_Q is the row hash threshold. Select the eigenvectors corresponding to n_2 eigenvalues to form the row projection of D_i, i.e. $Q = [\nu_1, \nu_2, \cdots, \nu_{n_2}]$. The size of Q is $q \times n_2$. P and Q respectively project D_i to obtain the principle component analysis matrix is $H = P \cdot D_i \cdot Q$. The size of H is $n_1 \times n_2$.

After the principal component analysis of the three-axis combined feature matrix, the dimensionality of the action feature matrix is reduced. This is conducive to the incremental learning of *farer*. Furthermore, in consideration of facilitating the construction of training samples, the matrix H after dimensionality reduction needs one-dimensional processing. We employ a flattening method to transform the matrix into a one-dimensional vector. After flattening, each one-dimensional vector matches the corresponding activity label as a training sample for the recognition model.

3.3 Recognition Model Construction and Incremental Update Based on BL

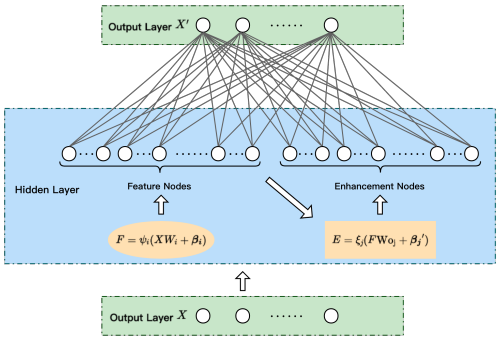

Fig. 2. Broad learning system model.

Broad learning system was proposed by Chen [24] in 2018. It is a structure of a single hidden layer including an input layer, a hidden layer and an output layer. Among them, the hidden layer includes a feature layer and an enhancement layer. First, the input layer receives the activity features which are extracted by windows. Then, the feature layer linearly maps activity features to construct feature nodes and the enhancement layer employs non-linear activation function for the feature nodes to obtain enhancement nodes. The feature nodes and the enhancement nodes are combined to form the hidden layer matrix. Finally, the output layer obtains the output coefficients by the method of pseudo inverse, and gives the learning results. The model architecture of BLS is shown in Figure 2.

Recognition Model Construction Based on BL. We adopt broad learning to build an activity recognition model. Suppose the training data sample set is $X = [x_1, x_2, \cdots, x_n]^T$ and each sample has m dimensions. The sample category label set is $C = [c_1, c_2, \cdots, c_n]^T$. The construction process of the recognition model is as follows:

1. Input layer.

Reduce the dimensionality of the action feature matrix and flatten it to obtain targeted one-dimensional data, which is employed as the training sample of the input layer.

2. Feature layer.

Suppose there are m_g groups of feature mapping and each group has n_g feature nodes. The i-th group of feature mapping is calculated as follows:

$$F_i = \psi_i(XW_i + \beta_i) \tag{3}$$

where $1 \le i \le m_g$, ψ_i is linear transformation function, W_i is a randomly generated matrix and β_i is a randomly generated vector. The integrated feature node matrix is $F = [F_1|F_2| \cdots |F_{m_g}]$.

3. Enhancement layer.

The main purpose of the enhancement layer is to increase the non-linear factor of the entire network. Since the feature nodes are all obtained in a linear manner, the BL recognition model introduces enhancement nodes to supplement it. Suppose there are m_e groups of enhancement nodes and each group has n_e enhancement nodes. The j-th group of enhancement nodes is calculated as follows:

$$E_j = \xi_j(FW_{o_j} + \beta_j) \tag{4}$$

where $1 \le j \le m_e$, ξ_i is nonlinear transformation function, W_{o_j} is a randomly generated orthogonal matrix and β_j is a randomly generated vector. The integrated feature node matrix is $E = [E_1|E_2| \cdots |E_{m_e}]$.

Finally, the feature node matrix F and the enhancement node matrix E are integrated to generate the hidden layer input matrix $\Lambda = [F|E]$.

4. Output layer.

The output layer mainly realizes the mapping from the input matrix of hidden layer to the label matrix. Since the category label matrix is C and the input matrix is Λ, if the mapping matrix Ω satisfies

$$\Lambda \cdot \Omega = C \tag{5}$$

then Ω can be obtained by matrix inversion. However, it should be noted that since Λ is generally not a square matrix, its pseudo-inverse can be solved as

$$\Lambda^{-1} = (\Lambda^T\Lambda + \delta I)^{-1}\Lambda^T \tag{6}$$

where I is the identity matrix, δ is the regularization coefficient and the value of δ is close to zero.

Incremental Update. When the recognition objects are some special users, such as lameness, large swing during the activity, etc., the false alarm rate of the model will increase, resulting in a poor user experience. Simply matching the activity characteristics of a special user to the recognition model makes it difficult to guarantee the recognition rate of the special activity. In addition, it also affects the accurate recognition of

the trained actions. On the other hand, if the special user's activity data is added to the source data and the model is retrained, a lot of model construction time is spent. To achieve targeted activity recognition, the recognition model needs to be updated incrementally.

By incrementally fusing the characteristics of user's behavior, we realize effective recognition of personalized activities. Our system *farer* has incremental learning capabilities and can be updated on the trained model without retraining historical data. This method satisfies activity recognition scenarios in more complex situations.

Denote $\dot{X}$ as the special activity data set and $\dot{C}$ as special activity label. The feature node matrix $F_{\dot{X}}$ and the enhancement node matrix $E_{\dot{X}}$ of $\dot{X}$ are given by the random matrix of *farer*. The hidden layer matrix is $\Lambda_{\dot{X}} = [F_{\dot{X}}|E_{\dot{X}}]$, and its pseudo-inverse is

$$\Lambda_{\dot{X}}{}^{-1} = [\Lambda^{-1} - \phi \cdot \sigma)|\phi] \tag{7}$$

where $\phi = \begin{cases} \omega^{-1} & \omega \neq 0 \\ (\Lambda^{-1}) \cdot \sigma \cdot (I + \sigma^T \cdot \sigma)^{-1} & \omega = 0 \end{cases}$, $\sigma = \Lambda_{\dot{X}} \cdot \Lambda^{-1}$, $\omega = \Lambda_{\dot{X}} - \sigma^T \cdot \Lambda$, I is the identity matrix.

After getting the incremental pseudo-inverse, the output of *farer* is calculated as

$$\dot{\Omega} = \Lambda_{\dot{X}}{}^{-1}\binom{C}{\dot{C}} = \Omega + \phi \cdot \left(\dot{C} - \Lambda_{\dot{X}} \cdot \Omega\right) \tag{8}$$

Therefore, the system in this paper can achieve rapid incremental update of the original model through matrix operations.

4 Experiment and Analysis

4.1 Experimental Settings

Table 1. Experimental parameter settings.

Parameter meaning	Value	Parameter
Smoothing filter window	51	μ
Polynomial order	3	k
Altitude threshold	0.5	Γ_A
Prominence threshold	1.2	Γ_P
Isolation threshold	55	Γ_I
Number of action windows	6	k_f
Column hash threshold	99.95%	∂_P
Row hash threshold	99.95%	∂_Q
Number of features	13	n_f
regularization coefficient	2^{-30}	δ
Number of feature windows	10	m_g
Number of feature nodes	12	n_g
Number of enhancement windows	1	m_e
Number of enhancement nodes	2000	n_e
Zoom scale	0.8	γ

To verify the recognition advantages of *farer*, we not only compare the classification effect of the traditional machine learning method SVM [25], but also compare the con-

volutional neural network CNN, LSTM, and their joint model CNN-LSTM and ConvLSTM.

We collect a large amount of three-axis accelerometer data of activities. There are 1556436 pieces of sampling data, and the sampling frequency is set to 180Hz. There are 7 activity states collected in the experiment, namely trickling, walking, brisk walking, jogging, upstairs, downstairs and jumping.

The action window is divided into 6 subsegments, employing typical features in the activity recognition research, which are the maximum, minimum, average, median, standard deviation, variance, interquartile range, skewness, kurtosis, root mean square, sum, range and entropy. The feature vector size of each coordinate axis is 1×65.

4.2 Performance Evaluation

System Recognition Accuracy. Table 2 shows the comparison of the activity recognition effect and model training time between *farer* and other methods. In the table, SA represents single activities and CA represents changed activities. According to the experimental results, the performance of *farer* is the best, with an overall recognition accuracy of 94.03%, which surpasses other recognition methods. Compared with single activity recognition, the accuracy of *farer* is 97.91%. More importantly, the recognition performance of *farer* is stable. It has high recognition accuracy for different activities, and there is no tendency deviation. In contrast, the recognition rate of other methods is either lower than that of *farer*, or has a higher false alarm rate for certain activities. For example, for downstairs, LSTM achieves 100% recognition accuracy, which exceeds 98.99% of *farer*. However, in the recognition of two more common activities, walking and brisk walking, the recognition rates of LSTM are only 73.98% and 78.20%, while the rates of *farer* are 96.64% and 97.78%.

Table 2. Comparison of the activity recognition effect.

		farer	**SVM**	**CNN**	**LSTM**	**CNN-LSTM**	**ConvLSTM**
SA	trickling	96.71%	96.58%	99.65%	97.90%	100.00%	100.00%
	walking	96.64%	93.33%	88.21%	73.98%	89.84%	96.34%
	brisk walking	97.78%	93.66%	93.23%	78.2%	92.48%	97.74%
	jogging	98.01%	98.94%	99.04%	99.05%	99.05%	98.1%
	upstairs	98.88%	98.23%	94.33%	98.38%	97.17%	94.33%
	downstairs	98.88%	98.09%	98.72%	100.00%	99.15%	99.15%
	jumping	100.00%	96.64%	98.29%	99.15%	100.00%	99.15%
	Overall	**97.91%**	96.4%	95.69%	92.33%	96.71%	97.74%
CA	Overall	**90.14%**	72.22%	76.67%	67.78%	81.11%	77.78%
SA/CA average		**94.03%**	84.31%	86.18%	80.06%	88.91%	87.76%

Comparing the situation of activity changes, the overall recognition rate of *farer* is 90.14%, which is much higher than other recognition methods. This is because other methods are affected by historical data. The historical data leads to a great reduction in recognition accuracy. The sliding window of *farer* only represents the current action, so it effectively avoids the interference of historical data. For example, the recognition accuracy of ConvLSTM which recognizes a single activity more accurately is reduced to 77.78%. LSTM's recognition of downstairs reaches 100%, but the recognition of

changed activities is only 67.78%. At the same time, the training time of these two types of models is much longer than *farer*.

Performance Comparison of Different Windows. The experiment in this section compares the three types of stable activities, activity changes slowly (S) and activity changes quickly (Q), as shown in Table 3. To verify the improvement of the activity recognition rate, we employ window data of different durations, namely 1.28s, 2.56s, 4s, 6s, 8s, and 10s. The sliding step of them is 1/2.

Table 3. Recognition accuracy of different sliding windows for activity scenes.

		Action window	**1.28*s***	**2.56*s***	**4*s***	**6*s***	**8*s***	**10*s***
Stable		97.91%	94.29%	95.41%	97.41%	97.12%	96.52%	97.91%
Changed	S	84.97%	65.71%	63.22%	65.46%	52.78%	37.04%	33.33%
	Q	95.75%	72.97%	69.57%	68.97%	60.53%	39.29%	31.82%

According to the experiment results, when the activity is stable, the recognition ability of action window is slightly higher than that of the fixed-size sliding window. However, when the activity changes, the advantage of the former increases significantly. As the window duration increases, the historical data has an increased influence on the feature extraction of the activity to be recognized. This results in a sharp drop in the recognition rate for fixed-size sliding windows.

When the activity changes slowly, such as trickling to walking, there is more interference data in the window because of the long switching time, resulting in a lower recognition rate. More importantly, the activity changes slowly, except for reasons of their own behavior habits, mostly because the front and back activities are similar. These changes further increase the difficulty of recognition. When the activity changes quickly, such as walking to upstairs, the difference between the front and back activities is generally large. As the activity changes quickly, the interference data in the window becomes less. The old activity is quite different from the new activity, so the recognition rate is increased.

Performance Comparison of Feature Matrix Dimensions. We adopt FETAD to extract features of the action data, and adopt C2DPCA to reduce the dimensionality of the three-axis feature matrix. According to the above parameter settings, the number of fine-grained subsegments is 6 and the number of features is 13. Thus, the number of features of each axis is $(6-1)\times 13 = 65$. The size of action feature matrix is 3×65. We compare different dimensionality reduction results, as shown in Figure 3.

Figure 3 shows the recognition accuracy of the three dimensionality reduction curves. The dimensionality of the feature matrix is reduced from 3×65 to $3 \times n$, $2 \times n$, and $1 \times n$, where $2 \le n \le 65$. Experimental results show that the recognition rate of $3 \times n$ is higher than that of the two categories. The experiment in this section further compares the difference between the two processing methods of flattening and square root of sum of squares (srss) when the row dimension is 3 after data dimensionality reduction.

Table 4 compares the highest recognition rate and the lowest recognition rate when the feature dimension drops to different sizes, as well as the corresponding row and column values. It can be seen from the table that the row dimension of 3 has the best

effect. Especially the lowest recognition rate is much higher than other cases. Considering the recognition accuracy and speed of *farer*, the column hash threshold and row hash threshold are both set to 99.95%. The size of the feature matrix after dimensionality reduction is 3×21. The recognition accuracy of the system is 97.91%.

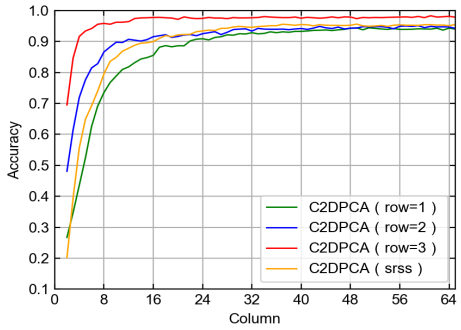

Fig. 3. Dimensionality reduction effect of *farer*.

Table 4. The maximum and minimum accuracy in the dimensionality reduction process.

C2DPCA	Row	Column	Accuracy (Max)	Column	Accuracy (Min)
flatten	1	51	94.39%	2	26.72%
flatten	2	59	95.08%	2	48.10%
flatten	3	61	98.28%	2	69.48%
srss	3	49	95.59%	2	20.22%

Performance Comparison of Incremental Update. Some users' behavior habits are different from most people's common activities. For traditional neural network, a lot of special behavior data need to be sampled as new training samples. These new training samples are appended to the original training samples. Then the model is retrained, which takes a long time. Even worse, it requires users to perform special activities for a long time to obtain adequate behavior samples. Obviously, this update method brings a lot of trouble to users and is impractical.

To effectively and quickly recognition special activities of different users, *farer* incrementally updates the recognition model. Based on the original model, it directly updates the model parameters according to the new activity data. The incremental update of *farer* greatly reduces users' activity sampling time and model training time, and ensures the balance of recognition accuracy. *farer* achieve a balance of the three.

We compare *farer* with traditional machine learning and deep learning from the perspective of recognition accuracy and model training time under the same sampling time. In the experiment, volunteers perform special behaviors and acts continuously for 30 minutes. In the sampling process, each learning model uses the activity data of this period to update the model at regular intervals. Among them, *farer* employs an incremental update method, while other systems mix the sampled data with the original training data and retrain models.

Figure 4 shows the recognition accuracy of special activities of different durations, with an interval of 5 minutes. It can be seen from the figure that the recognition accuracy of this system is the highest when the sampling time is the same. *farer* is growing

faster than other methods.

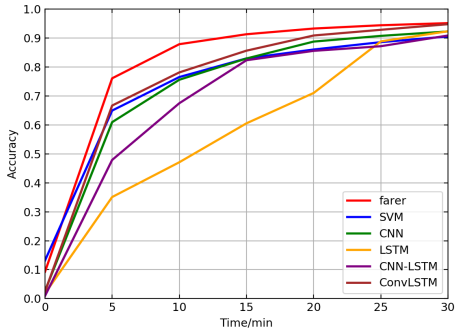

Fig. 4. Recognition accuracy of special activities with different durations

5 Conclusion

By studying the problem of poor accuracy of activity recognition for sliding windows, we propose a fine-grained activity recognition method, which employs fine-grained subsegments and incremental broad learning. We also design the corresponding activity recognition system *farer*. The system can effectively process the activity data, realize the accurate segmentation of activities and ensure the effectiveness of the action feature extraction. Furthermore, the fine-grained subsegment division is used to dig deeply into the action features and reduce the dimensionality to ensure the recognition rate of the activity. In the process of activity recognition, the incremental recognition model based on broad learning is adopted to learn the activity behavior of special users, which improves the user experience. After a large number of experiments, the performance of *farer* is better than that of other recognition methods. The recognition rates for stable activities and changed activities are 97.91% and 90.14%. Particularly, the activities often change during actual applications. *farer* has a recognition accuracy far exceeding other methods for this situation. Meanwhile, our system requires only a small amount of special activity data, and after a short period of training, it achieves a high recognition rate. Therefore, *farer* can face more complex situations and has good practicability.

Reference

1. Jiang W, Miao C, Ma F, et al. Towards environment independent device free human activity recognition[C]//Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. 2018: 289-304.
2. Youke Wu, Haiyang Huang, Ningyun Wu, Yue Wang, Md Zakirul Alam Bhuiyan and Tian Wang, An Incentive-Based Protection and Recovery Strategy for Secure Big Data in Social Networks, Information Sciences, 2020, 508: 79-91.
3. L. Zhao, W. Zhao, A. Hawbani, A. Al-Dubai, G. Min, A. Y. Zomaya, C. Gong, “Novel Online Sequential Learning-based Adaptive Routing for Edge Software-Defined Vehicular Networks,” IEEE Transactions on Wireless Communications, 2020.
4. L. Zhao, G. Han, Z. Li, L. Shu, “Intelligent Digital Twin-based Software-Defined Vehicular Networks”, IEEE Network, 2020.
5. L. Zhao, H. Li, N. Lin, M. Lin, C. Fan, J. Shi, “Intelligent Content Caching Strategy in Autonomous Driving

Towards 6G," IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2021.

6. Tian Wang, Qun Wu, Sheng Wen, Yiqiao Cai, Hui Tian, Yonghong Chen, Baowei Wang. Propagation modeling and defending of a mobile sensor worm in wireless sensor and actuator networks. Sensors, 2017, 17(1): 139.
7. Tian Wang, Hao Luo, Xiangxiang Zeng, Zhiyong Yu, Anfeng Liu, Arun Kumar Sangaiah. Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(3): 1797-1806.
8. J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, ‘‘Deep learning for sensorbased activity recognition: A survey,’’ Pattern Recognit. Lett., vol. 119, pp. 3–11, Mar. 2019.
9. Yang J, Bang W, Choi E, et al. A 3D Hand-drawn Gesture Input Device Using Fuzzy ARTMAP-based Recognizer[J]. Journal of Systemics，Cybernetics and Informatics, 2006, 4(3):1-7.
10. E. Garcia-Ceja and R. Brena, Long-term activity recognition from accelerometer data, Procedia Technology, vol. 7, pp. 248-256, 2013.
11. Khan A M , Lee Y K , Lee S , et al. Accelerometer's position independent physical activity recognition system for long-term activity monitoring in the elderly[J]. Medical & Biological Engineering & Computing, 2010, 48(12):1271-1279.
12. B. Fida, I. Bernabucci, D. Bibbo, S. Conforto, and M. Schmid, ‘‘Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer,’’ Med. Eng. Phys., vol. 37, no. 7, pp. 705–711, Jul. 2015.
13. Elsts A , Twomey N , Mcconville R , et al. Energy-efficient activity recognition framework using wearable accelerometers[J]. Journal of Network and Computer Applications, 2020, 168:102770.
14. M. M. Hossain Shuvo, N. Ahmed, K. Nouduri and K. Palaniappan, "A Hybrid Approach for Human Activity Recognition with Support Vector Machine and 1D Convolutional Neural Network," 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2020, pp. 1-5.
15. K. Xia, J. Huang and H. Wang, "LSTM-CNN Architecture for Human Activity Recognition," in IEEE Access, vol. 8, pp. 56855-56866, 2020.
16. Cha, S. H., Seo, J., Baek, S. H., & Koo, C. (2018). Towards a well-planned, activity-based work environment: Automated recognition of office activities using accelerometers. Building and Environment, 144, 86-93.
17. S. W. Pienaar and R. Malekian, "Human Activity Recognition using LSTM-RNN Deep Neural Network Architecture," 2019 IEEE 2nd Wireless Africa Conference (WAC), 2019, pp. 1-5.
18. Gao W, Zhang L, Teng Q, et al. DanHAR: Dual attention network for multimodal human activity recognition using wearable sensors[J]. Applied Soft Computing, 2021, 111: 107728.
19. M. Panwar, S. R. Dyuthi, K. C. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G. R. Naik, ‘‘CNN based approach for activity recognition using a wrist-worn accelerometer,’’ in Proc. EMBC, Seogwipo, South Korea, Jul. 2017, pp. 2438–2441.
20. Song-Mi Lee, Sang Min Yoon and Heeryon Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), 2017.
21. R. Mutegeki and D. S. Han, "A CNN-LSTM Approach to Human Activity Recognition," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2020, pp. 362-366.
22. F. J. Ordóñez and D. Roggen, ‘‘Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,’’ Sensors, vol. 16, no. 1, p. 115, Jan. 2016.
23. H. Chen et al., "Assessing impacts of data volume and data set balance in using deep learning approach to human activity recognition," 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2017, pp. 1160-1165.
24. Fu Z, He X, Wang E, et al. Personalized Human Activity Recognition Based on Integrated Wearable Sensor and Transfer Learning[J]. Sensors, 2021, 21(3): 885.
25. Hong J H, Ramos J, Dey A K. Toward personalized activity recognition systems with a semipopulation approach[J]. IEEE Transactions on Human-Machine Systems, 2015, 46(1): 101-112.