Hindawi
Security and Communication Networks
Volume 2020, Article ID 2132138, 12 pages
https://doi.org/10.1155/2020/2132138

Research Article

Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques

Huaijun Wang,[1,2] Jing Zhao,[1] Junhuai Li,[1,2] Ling Tian,[1] Pengjia Tu,[1] Ting Cao,[1,2] Yang An,[1] Kan Wang,[1,2] and Shancang Li[3]

[1]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*
[2]*Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China*
[3]*Department of Computer Science and Creative Technologies, UWE Bristol, Bristol BS16 1QY, UK*

Correspondence should be addressed to Junhuai Li; lijunhuai@xaut.edu.cn

Received 16 February 2020; Revised 8 June 2020; Accepted 6 July 2020; Published 27 July 2020

Academic Editor: Xiaolong Xu

Human activity recognition (HAR) can be exploited to great benefits in many applications, including elder care, health care, rehabilitation, entertainment, and monitoring. Many existing techniques, such as deep learning, have been developed for specific activity recognition, but little for the recognition of the transitions between activities. This work proposes a deep learning based scheme that can recognize both specific activities and the transitions between two different activities of short duration and low frequency for health care applications. In this work, we first build a deep convolutional neural network (CNN) for extracting features from the data collected by sensors. Then, the long short-term memory (LTSM) network is used to capture long-term dependencies between two actions to further improve the HAR identification rate. By combing CNN and LSTM, a wearable sensor based model is proposed that can accurately recognize activities and their transitions. The experimental results show that the proposed approach can help improve the recognition rate up to 95.87% and the recognition rate for transitions higher than 80%, which are better than those of most existing similar models over the open HAPT dataset.

1. Introduction

Human behavior recognition (HAR) is the detection, interpretation, and recognition of human behaviors, which can use smart heath care to actively assist users according to their needs. Human behavior recognition has wide application prospects, such as monitoring in smart homes, sports, game controls, health care, elderly patients care, bad habits detection, and identification. It plays a significant role in depth study [1] and can make our daily life become smarter, safer, and more convenient.

Currently, human behavior data can be acquired in two ways: one is based on computer vision and the other is based on sensors [2]. Behavior recognition based on computer vision has been studied for a long time and has a mature theoretical basis. However, the vision-based approaches have many limitations in practice. For example, the use of a camera is limited by various factors, such as light, position, angle, potential obstacles, and privacy invasion issues, which make it difficult to be restricted in practical application. Although the research time of sensor-based behavior recognition is relatively short, with the development and maturity of microelectronics and sensor technology, there are various types of sensors, such as accelerometers, gyroscopes, magnetometers, and barometers. These sensors can be integrated into mobile phones and wearable devices such as watches, bracelets, and clothes. Furthermore, state-of-the-art wearable sensors have solved the issue of antimagnetic field interference, such as [3], which can accurately estimate the current acceleration and angular velocity of motion sensors in real time in the presence of magnetic field interference. So these wearable sensors are usually small in size, high in sensitivity, and strong in anti-interference ability, so the sensor-based identification method is more suitable for practical situations. Moreover, sensor-based behavior recognition is not limited by scene or time, which

can better reflect the nature of human activities. Therefore, the research and application of human behavior recognition based on sensors are more and more valuable and significant.

Besides, the HAR includes two types: basic actions and transition actions. Due to the low incidence and short duration of transition movement, there are relatively few studies on the transition movement from standing to sitting, walking to standing, and so on in the research of human behavior recognition [4]. However, the study of transitional movement is a very important part of human behavior recognition. In order to improve the behavior recognition rate, transition action recognition is not negligible. The transition action is the distinction of a variety of basic actions in frequent alternations. The accurate division of the transition action can accurately segment the streaming data to a certain extent and ultimately improve the recognition rate. In addition, the behavior recognition methods based on traditional patterns have shortcomings such as manual feature extraction. With the application and development of deep learning in different fields, the deep learning model also shows great advantages in the field of behavior recognition.

The main contributions of this work are summarized as follows:

(1) We presented a deep learning model composed of convolutional and Long Short-Term Memory recurrent layers, which can automatically learn local features and model the time dependence between features.

(2) We discussed the influence of key parameters in deep learning model on performance and finally determined the best parameters in the model.

(3) We analyzed and compared the experimental results with other models that adopt the same common data set. The results show that the proposed method is superior to the other advanced methods.

In this work, we use both acceleration sensor and a gyroscope sensor of smart phones to acquire data and proposed a CNN-LSTM hybrid model to recognize the transition motion. Convolution neural network (CNN) [5] is a type of depth neural network used as a feature extractor. It is characterized by local dependence, so it has good performance in extracting local features. However, human activity information belongs to long instance, which is composed of complex movements and changes with time. So the CNN model does not work well in extracting the relationship between time and features. The Long Short-Term Memory (LSTM) [6] neural network is a kind of recursion network that contains a memory to simulate a time dependent sequence problem. Therefore, the mixture of CNN-LSTM can accurately identify the basic and transitional features of activities.

The remainder of the paper is organized as follows: Section 2 reviews the literature on human activity identification based on deep learning and existing problems; Section 3 presents the mixed deep learning framework proposed in this paper for existing problems; Section 4 discusses and analyzes the experimental results based on experimental data. Finally, Section 5 concludes this paper.

2. Related Works

Due to the extensive application of human-computer interaction, behavior detection, and other technologies, human behavior recognition has become a hot field [7]. Human behavior recognition can be regarded as a representative pattern recognition problem. The traditional pattern of behavior recognition research using decision tree, support vector machine (SVM), and other machine learning algorithms can obtain much satisfactory results, in premise of some controlled experimental environments and a small number of labeled data. However, the accuracy of these methods depends on the effectiveness and comprehensiveness of manual feature extraction. In addition, these methods can only extract shallow features. Because of these limitations, the behavior recognition methods based on traditional pattern recognition are limited in classification accuracy and model generalization.

In recent years, deep learning has developed rapidly and attracted many research efforts, especially in image, processing time series, natural language, logical reasoning, and other complex data processing aspects and has achieved unparalleled achievements [8]. Different from the traditional behavior recognition method, deep learning could reduce the workload of feature design. In addition, the higher-level and more meaningful features can be learned via the end-to-end neural network. Furthermore, the deep network structure is more suitable for unsupervised incremental learning. Moreover, deep networks created by superimposing several layers of features can model data with complex structures. In a word, the deep learning is an ideal method for HAR.

Since deep learning has made outstanding achievements in image feature extraction, many researchers first try to apply it to behavior recognition based on video. In early periods, Taylor et al. [9] used convolution threshold *Boltzmann* machine to identify video behavior data and extract sensitive features. Ji et al. [10] proposed a three-dimensional CNN model to capture more action information from space and time. Liu et al. [11] proposed that CNN and conditional random domains (CRFs) be combined for action segmentation and recognition. The CNN can automatically learn space-time characteristics, while CRF is able to capture the dependency between outputs. Other common deep learning methods are also widely used, such as recursive neural network [12] and long short-term memory network. On one hand, it is successful on application of deep learning in video behavior recognition. On the other hand, it is also widely used in human behavior recognition based on sensors.

Zeng et al. [13] proposed treating the single-axis sensor data as one-dimensional data of images and then sending them to CNN for identification. Jiang and Yin [14] combined the signal sequences of accelerometer and gyroscope into an active image, enabling deep convolutional neural network (DCNN) to automatically learn the optimal features

from the active image. Chen and Xue [15] modified the CNN convolution kernel to adapt to the characteristics of triaxial acceleration signals. Ronao and Cho [16] proposed a convNet, which realized efficient and data adaptive human behavior recognition with smart phone sensors. ConvNets not only utilize the inherent time-local dependence of sensor signal sequences but also provide an adaptive method for extracting robust features. Experimental results show that this method can recognize similar actions, which are difficult to be processed by traditional machine learning. Murad and Pyun [17] and Zhou et al. [18] proposed three deep recursive neural network structures based on LSTM to establish recognition models to capture time relations in input sequences and could achieve more accurate recognition. Due to the superior performance of LSTM in behavior recognition application, Guan and Plötz [19] and Qi et al. [20] improved the LSTM and proposed an integration model, integrating different LSTM learners into an integrated classifier. Through the experimental evaluation in the standard data set, it is proved that the integrated system composed of LSTM learners is superior to a single LSTM network. Ignatov [21] combined the manually extracted statistical features with the features automatically extracted by neural network and realized a human behavior recognition method based on user autonomous deep learning. Among them, CNN extracted local features, while statistical features preserved the information about the global form of time series. Experiments on open data sets show that the model has the advantages of small computation, short running time, and good performance. Nweke et al. [22] and Wang et al. [23], respectively, summarized the application of deep learning method in sensor-based behavior recognition and not only put forward detailed views on the existing work, but also pointed out the challenges and improvement directions of future research.

This work demonstrated the potential of deep neural network to learn the potential features and time series features. Nevertheless, existing works on action recognition mainly focus on the aspect of basic behavior recognition, while the transition between actions is usually ignored because the transition action has a short duration. However, it is necessary to study the transition action in depth in order to improve the robustness of the model. The precise division of the transition action can accurately segment the streaming data to a certain extent and ultimately improve the recognition rate. In this paper, CNN combined with LSTM hybrid model is adopted to extract deep and advanced features, and elaborate description is made of basic and transition action, so as to realize accurate identification.

3. Proposed Method

The overall architecture diagram of the method proposed in this paper is shown in Figure 1, which contains three parts. The first part is the preprocessing and transformation of the original data, which combines the original data such as acceleration and gyroscope into an image-like two-dimensional array. The second part is to input the composite image into a three-layer CNN network that can automatically extract the motion features from the activity image and abstract the features, then map them into the feature map. The third part is to input the feature vector into the LSTM model, establish a relationship between time and action sequence, and finally introduce the full connection layer to achieve the fusion of multiple features. In addition, Batch Normalization (BN) is introduced [24], in which BN can normalize the data in each layer and finally send it to the S*oftmax* layer for action classification.

3.1. Data Preprocessing. Due to the large amount of behavioral data collected by the sensor, it is impossible to input all the data into the depth model at one time. Therefore, sliding window segmentation should be carried out before data input into the model. The behavior recognition method proposed in this paper can recognize both the basic action and the transition action at the same time. The transition action lasts for a short time; it is necessary to choose the appropriate window size. If the window is too large, important information will be lost. Otherwise, the computational costs will be increased. After data segmentation, the behavioral data collected by sensors are one-dimensional time series different from image data. Therefore, before applying the deep learning model to these input data, it is necessary to input and adapt them. Dimension transformation is carried out on the data after window segmentation. The method of transformation is to splice the sensor data of all axes into a two-dimensional matrix. The advantage of this approach to data processing is that it preserves the correlation between sensors' axes. Finally, samples similar to pictures are formed and input into the deep learning model. Figure 2 shows the model structure of data preprocessing.

3.2. Feature Learning Based 1D-CNN. The original uniaxial acceleration and gyroscope data are equivalent to two-dimensional array of images after dimensional transformation. The feature image is input into the convolution neural network, and its structure is generally composed of convolution layer and pooling layer. The convolution layer carries out convolution operation on the input image through convolution kernel to obtain feature mapping. The pooling layer extracts local features from the feature map of the convolution layer through sampling operation to lessen the size of neurons and the number of parameters. The convolution layer and pooling layer are stacked to form a deep structure, which can automatically extract the action feature information from the original action data [5].

The CNN model structure designed in this paper is shown in Figure 3. The CNN network model consists of three convolution layers and three pooling layers (each convolution layer is followed by one pooling layer) and finally outputs a number of feature map images with action features. Table 1 illustrates the settings of different parameters for each convolution and pooling layer. Convolution is achieved by the convolution of two-dimensional convolution kernel with images superimposed by multiple adjacent frames. The convolution kernel number of the three convolution layers is 18, 36, and 72, respectively. The

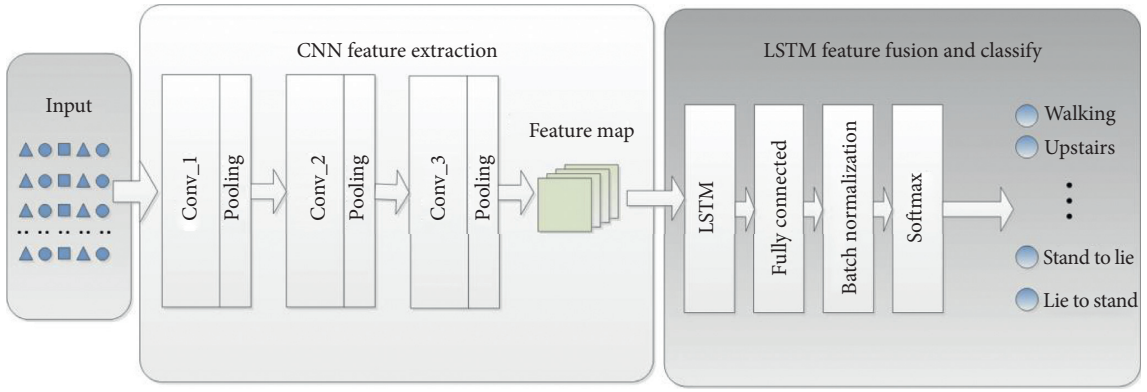

Figure 1: Human activity recognition framework based on CNN-LSTM.

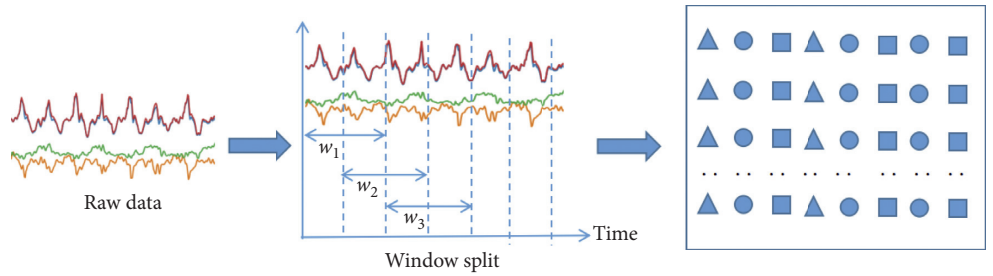

Figure 2: Structure of data preprocessing model.

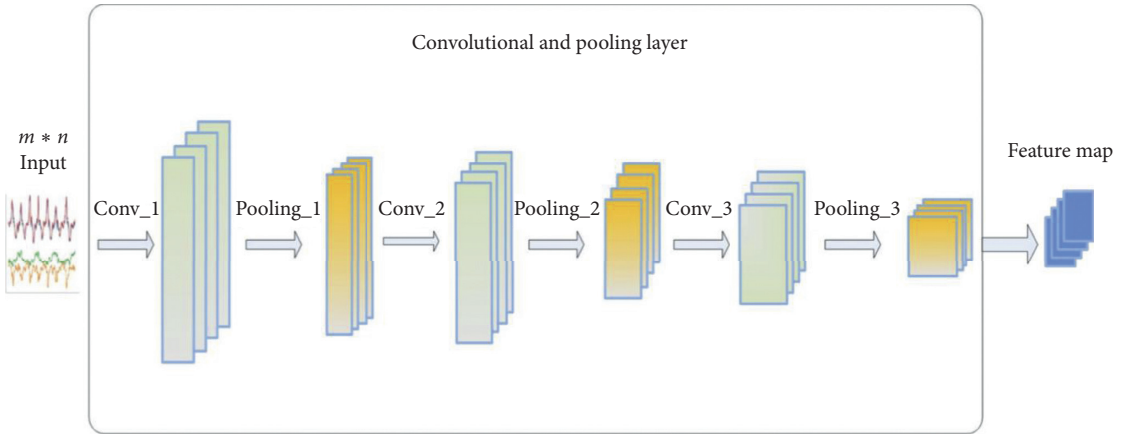

Figure 3: CNN model architecture.

Table 1: Activity label corresponding to the original data.

Id	Exp	Label	Start	End
1	1	5	250	1232
1	1	7	1233	1392
1	1	4	1393	2194
1	1	8	2195	2359
1	1	5	2360	3374
1	1	11	3375	3662
1	1	5	3663	4538
1	1	11	4539	4735
1	1	5	4736	5667
1	1	11	5668	5859
1	1	5	5860	6786
1	1	11	6787	6977
1	1	5	6978	8078

convolution kernel size is 2×8, 2×18, and 2×36, and the step size is 1. Since the filter may not be able to process the data in a certain direction in the operation of convolution, to avoid reducing data of the image edge, the padding parameter is introduced and set to "SAME" and 0 is added to the edge of the input image matrix. After the convolution operation in the convolution layer, the output will usually pass through a nonlinear activation function and then form the output of the convolution layer. The popular activation functions include *Sigmoid* function, *ReLU* function, and Tanh function. Among them, *ReLU* function can change the negative value of the data extracted by CNN into 0, and the positive value of the data greater than 0 remains unchanged. After nonlinear processing operation, the positive value

greater than 0 can be more clearly expressed by the extracted features. Therefore, *ReLU* activation function is used in the convolution layer of CNN:

$$f(x) = \max(0, x) = \begin{cases} 0, & x < 0, \\ x, & x \geq 0. \end{cases} \tag{1}$$

Further, we have

$$f'(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases} \tag{2}$$

Pooling layer is regarded as reducing the number of feature mappings and parameters. The popular pooling techniques include maximum pooling and average pooling. In recent years, relevant theoretical analysis and performance evaluation have shown the superior performance of the maximum pooling strategy, which is widely used in deep learning [25, 26]. Moreover, some studies show that the maximum pooling technology is very suitable for sensor-based human behavior recognition [27]. Therefore, all pooling layers of CNN in this paper utilized the maximum pooling technique. Specific convolution and pooling process parameters are set as shown in Table 2.

3.3. Feature Fusion and Action Classification. To improve the recognition rate of transition actions, we build a LSTM after the CNN network $\{f_1, f_2 \ldots\ldots f_n\}$ is the feature sequence converted from the feature map calculated by CNN from the images composed of original data. Therefore, the sequence $\{f_1, f_2 \ldots\ldots f_n\}$ input LSTM and the storage unit of LSTM will produce a sequence of characters $\{m_1, m_2 \ldots\ldots m_n\}$.

Since LSTM has different gating units, memory units such as input gate, forgetting gate, and output gate are combined with learning weights to solve the problem of gradient disappearance in the process of back propagation of ordinary circular neural network. Meanwhile, LSTM can model time-dependent actions and fully capture global features, so as to improve the recognition accuracy [28]. LSTM cell controls the inward flowing information of neurons, which is composed of forgetting gate, input gate, and output gate. Furthermore, the predicted value of LSTM cell is obtained using Tanh function.

Firstly, the forgetting gate determines how much information from the previous moment can be accumulated to the current cell. As shown in equation (3), the probability value is calculated to determine the amount of information that can pass through the gate:

$$\Gamma_f = \sigma\left(w_f * \left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_f\right), \tag{3}$$

where w_f represents the weight corresponding to the input vector, b represents the bias, $a^{\langle t-1\rangle}$ presents the output of the neuron at the last moment, and $x^{\langle t\rangle}$ represents the current input of the neuron.

Secondly, the input gate consists of update gate and Tanh layer, which controls how much information can flow into the current cell. The calculation process is shown in equations (4)–(6). The input of the input gate and the output of the forgetting gate update the cell at the same time, discarding unwanted information. Then, the predicted value of the current unit is determined by the output gate, and the output of the model is obtained, as shown in equations (7) and (8):

$$\Gamma_u = \sigma\left(w_u * \left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_u\right), \tag{4}$$

$$\widetilde{C} = \tanh\left(w_c * \left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_c\right), \tag{5}$$

$$C_t = \Gamma_u * \widetilde{C}^{\langle t\rangle} + \Gamma_f * C^{\langle t-1\rangle}, \tag{6}$$

$$\Gamma_o = \sigma\left(w_o * \left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_o\right), \tag{7}$$

$$a^{\langle t\rangle} = \Gamma_o * \quad \tanh\left(C^{\langle t\rangle}\right). \tag{8}$$

After the processing of LSTM layer, the final output is a set of vectors containing time and action sequence correlation, which are input into the full connection layer for the fusion of global action features. The training process of neural network model becomes complicated since the statistical distribution of input of each layer changes with the parameters of the previous layer. To keep the distribution of output data from changing too much, a lower learning rate will be used, which could reduce the training speed. To solve this issue, this paper introduces the BN to standardize the values of each layer in LSTM (the output of neurons at the last moment and the input at the current moment), so that the mean and variance of sum will not change with the change of the distribution of the underlying parameters and effectively separate the parameters of each layer from other layers. In this way, the gradient disappearance or explosion can be prevented and the training speed of the network can be accelerated. The BN algorithm is shown in Algorithm 1.

In Algorithm 1, μ_x and ς_x^2 are the mean and variance of x_i obtained through minibatch. The mean and variance were used to normalize x_i to make the sample follow normal distribution. However, the positive distribution is not able to reflect the characteristic distribution of the training samples, and thus it is necessary to introduce the scaling factor γ and the shift factor β. As training progresses, γ and β are also learned by back propagation to improve accuracy.

After BN operation, the features are more obvious, so input them to *Softmax* layer to extract the action features and classify them in time series. In this model, the output layer uses *Softmax* normalized exponential function to calculate the posterior probabilities of different actions to

Table 2: The convolution and pooling layers of the CNN architecture.

Layers	Conv1d_1	Conv1d_2	Conv1d_2
Size	$1 \times 2 \times 8$	$1 \times 2 \times 18$	$1 \times 2 \times 36$
Stride	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$1 \times 1 \times 1$
Channel	18	36	72
Layers	Pooling_1	Pooling_2	Pooling_3
Size	$1 \times 2 \times 18$	$1 \times 2 \times 36$	$1 \times 2 \times 72$
Stride	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$1 \times 1 \times 1$
Channel	18	36	72

Input: data set: $\chi = \{x_1 \ldots x_n\}$
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$
(1) Calculate the mean of data set: $\mu_x \leftarrow (1/n)\sum_{i=1}^{n} x_i$
(2) Calculate the variance of data set: $\varsigma_\chi^2 \leftarrow (1/n)\sum_{i=1}^{n}(x_i - \mu_\chi)^2$
(3) Normalize data: $\widehat{x}_i \leftarrow (x_i - \mu_x/\sqrt{\varsigma_x^2 + \varepsilon})$
(4) Scale change and deviation: $y_i \leftarrow \gamma\widehat{x}_i + \beta = \mathrm{BN}_{\gamma,\beta}(x_i)$
(5) Return learning parameter γ and β

Algorithm 1: Algorithm of batch normalization.

realize classification. It maps the output values of neurons between (0, 1), which can be regarded as the prediction probability of actions, and the largest one is the result of classification. Then the *Softmax* output layer outputs a category vector such as [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0], indicating that the classification result is an action numbered 5.

3.4. Model Implementation and Training. The neural network described here is implemented in TensorFlow [29]. It is a lightweight library for building and training neural networks. Model training and classification runs on a conventional computer with a 2.4 GHz CPU and 16 GB memory.

The model is trained in a fully supervised manner to backpropagate the gradient from the *Softmax* layer to the convolution layer. Network parameters are optimized by using minibatch gradient descent method and Adam optimizer through minimizing cross-loss function [13]. Adam is widely used due to its advantages in simple implementation, efficient calculation, and low memory demand. Compared with other kinds of random optimization algorithms, Adam has great advantages. In this paper, to better train the model, after the training data are input into the network. Adam optimizer and backpropagation algorithm are used to learn and optimize the network parameters. Meanwhile, the cross-entropy loss function is used to calculate the total error, as shown in the following equation:

$$C = -\frac{1}{N}\sum_{X}[y\mathrm{In}a + (1-y)\mathrm{In}(1-a)], \tag{9}$$

where y is the true tag and a is the predicted value.

To improve efficiency, small batches of data segment size are segmented during training and testing. With these configurations, the cumulative gradient of the parameters is calculated after each small batch. The weights are randomly and orthogonally initialized. As a form of regularization, we introduce a dropout operator on each dense layer of input. This operator sets the activation of a randomly selected unit to zero during training. Dropout technology proposed by Hinton et al. [30] is based on the principle of randomly deleting some nodes in the network while maintaining the integrity of input and output neurons, which is equivalent to training many different networks. Different networks may overfit in different ways, but their average results can effectively reduce overfitting. In addition, dropout allows neurons to learn stronger features by not relying on other specific neurons. The number of parameters to be optimized in a deep neural network varies depending on the type of layer it contains. And it has a great impact on the time and computer skills required to train the network. The specific model training parameters will reflect the best choices in the experiment.

4. Activity Recognition

4.1. Experiment Data. In addition to common basic actions, this paper also studies transition actions. Actually, a few existing public data sets contain transition actions. Therefore, this paper adopts the international standard Data Set, Smart phone Based Recognition of Human Activities and Postural Transitions Data Set [31, 32] to conduct an experiment, which is abbreviated as HAPT Data Set. The data set is an updated version of the UCI Human Activity Recognition Using popularity Data set [8]. It provides raw data from smart phone sensors rather than preprocessed data. In addition, the action category has been expanded to include transition actions. The HAPT data set contains twelve types of actions. Firstly, it has six basic actions that include three types of static actions, such as standing, sitting, and lying, and three types of walking activities such as walking, going downstairs, and upstairs; Secondly, it has six possible transitions between any two static movements: standing to sitting, sitting to standing, standing to lying, lying to sitting, sitting to lying, and lying to standing.

The HAPT data collection process is shown in Figure 4. The experiment involved 30 volunteers, whose ages range from 19 to 48, each wearing a smart phone on their waist. Data collection is carried out with the built-in acceleration sensor and gyroscope, and the sampling frequency is 50 Hz. Meanwhile, video records of the experimental process are made for the convenience of subsequent data marking.

The collected data is saved in the form of .txt, and the acceleration and gyroscope data are stored independently, with 60 groups, respectively. As shown in Table 1, it is the label information corresponding to the original data of the experiment. Among them, the first column is the experiment ID, the second column is the experimenter number, the third column is the action label, and the fourth and fifth columns are the start and end row labels of the corresponding sensor data. The label ranges from 1 to 12, representing 12 types of actions. It can be seen from the figure that the collected data contains invalid data, and the first 250 pieces of data are unlabeled and belong to invalid data.

Figure 4: Data collection of the physical activities.

Table 3: The data amount of various activities in the HAPT.

Type	ID	Number
Walk	*A*1	122,091
Upstairs	*A*2	116,707
Downstairs	*A*3	107,961
Sit down	*A*4	126,677
Stand	*A*5	138,105
Lie	*A*6	136,865
Stand to sit	*A*7	10,316
Sit to stand	*A*8	8,029
Sit to lie	*A*9	12,428
Lie to sit	*A*10	11,150
Stand to lie	*A*11	14,418
Lie to stop	*A*12	10,867

After preliminary processing of the original data, all the data without labels were deleted. Finally, 815,614 valid pieces of data were obtained. Due to the low frequency and short duration of transition action, as well as the high frequency and long duration of basic action, there is a considerable difference in data volume between transition action and basic action. The data volume of the six transition actions is much lower than that of the other basic actions, accounting for only about 8% of the total data. Table 3 lists the amount of data for different actions. The original data is divided into three parts, training set, verification set, and test set, in which the training set is used for model training, and verification set is used to adjust parameters, and test set is used to measure the quality of the final model.

4.2. Parameters Setting. In the deep learning network, the model parameters greatly affect its recognition rate. Therefore, the experimental analysis of the number of neurons, learning rate, BN, Batch size, and other parameters in LSTM layer would be conducted in the following sections.

4.2.1. Number of Neurons in LSTM Layer. In order to verify the influence of the number of neurons in LSTM layer on the recognition results, the following experiments are carried out in this paper, as shown in Figure 5. It shows that the recognition rate is the lowest when each LSTM layer contains only 8 neurons. This is because, given less neurons, the network lacks the necessary learning ability and information processing ability, resulting in the low recognition rate. As

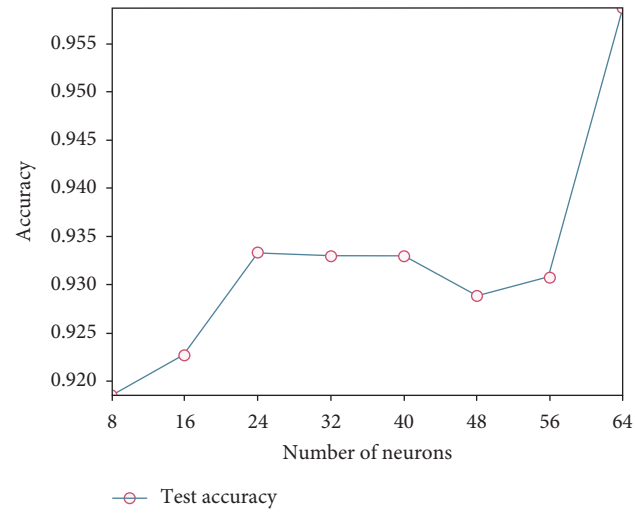

Figure 5: Accuracy of different numbers of neurons on test sets.

the number of neurons increases, the recognition rate tends to increase. When the number of neurons is 64, the recognition rate reaches 95.87%. If the number of neurons is too large, the complexity of network structure will increase and the learning speed of network will slow down. Therefore, considering the training time of the network, the number of LSTM layer neurons in this paper is tentatively 64.

4.2.2. The Learning Rates. Experiments are carried out at different learning rates in this paper. As shown in Table 4, it can be seen that the recognition rate of the model reaches a maximum of 95.87% when the learning rate is 0.002. Therefore, the learning rate of 0.002 is adopted.

4.2.3. BN Operation. To verify the improvement of the BN operation on the network model, a comparative experiment is carried out first with and without BN layer. The epoch is set to 400, and other parameters remain unchanged. The recognition rates of both methods on the test set are shown in Table 5. Obviously, the recognition rate on the test set is improved by about 4.24% after the BN layer is added.

4.2.4. Batch Size. Batch size refers to the Batch sample size, whose maximum value is the total number of samples in the

Table 4: Accuracy of different learning rates on test sets.

Learning rate	Recognition rate (%)
0.001	93.57
0.0015	94.21
0.002	95.87
0.0025	92.39
0.003	93.34
0.0035	92.12
0.004	92.84
0.0045	92.01

Table 5: Accuracy and loss rate on test sets with or without BN layer.

	Recognition rate (%)
Without BN layer	91.63
With BN layer	95.87

Table 6: Accuracy of different batch size on test sets.

Batch size	Recognition rate (%)
25	91.74
50	92.88
75	92.92
100	93.10
125	94.33
150	95.87
175	93.45
200	93.37
225	93.72
250	93.45
275	92.84
300	93.35
325	94.06
350	93.34
375	92.96
400	93.53

training set. When the amount of data is small, the batch data is the whole data set, so that it can approach the extreme value direction more accurately. However, in practical applications, the amount of data used by deep learning is relatively large, and the principle of small batch processing is generally adopted. Using small batch processing requires relatively little memory and faster training time. Within an appropriate range, increasing the batch size can more accurately determine the direction of gradient descent and cause less training shock. However, when the batch size increases to a certain value, the determined downward direction will not change and the correction of parameters will slow down significantly. The identification results of different batch sizes are shown in Table 6. It can be seen that when the batch size is 150, the maximum identification rate reaches 95.87%. Therefore, 150 is selected as the best batch size in this paper.

The parameters of the CNN-LSTM model proposed in this paper are shown in Table 7.

Table 7: Experimental parameters of CNN-LSTM model.

Parameters	Value
Input vector size	150
Input channel number	8
Convolution kernel size	3
Pool size	2
Activation function	ReLu
LSTM layer	1
Neurons number	64
Dropout	0.5
Learning rate	0.002
Batch size	150
Epoch	400

5. Experimental Results and Analysis

For human movement recognition, Wang and Liu [33] proposed to use the F-measure standard measurement method to verify the performance of the deep-rooted LSTM network model in human activity recognition. Lu et al. [34] demonstrated the superiority of the model in behavior recognition by using accuracy, prediction rate, and recall rate in the experiment. Therefore, to evaluate the performance of the motion recognition method proposed in this paper, we also used the measurement method of accuracy, recall rate, loss rate, and F-measure in the experiment.

According to the above parameters, the recognition confusion matrix of 12 different actions is shown in Table 8. Accuracy curve of CNN-LSTM model is shown in Figure 6. It can be seen from Table 9 that the overall recognition rate of CNN-LSTM is high, and the CNN-LSTM has a better recognition effect on the transition action.

Table 8: Confusion matrix of various actions.

Actual	Predict											
	*A*1	*A*2	*A*3	*A*4	*A*5	*A*6	*A*7	*A*8	*A*9	*A*10	*A*11	*A*12
*A*1	410	1	3	0	0	0	0	0	0	0	0	0
*A*2	5	388	3	0	0	0	0	0	0	0	0	0
*A*3	1	3	346	0	0	0	0	0	0	0	0	0
*A*4	1	0	0	383	32	3	1	0	1	1	0	0
*A*5	0	0	1	31	431	0	0	0	0	0	0	0
*A*6	0	0	0	1	0	457	0	0	0	0	0	0
*A*7	0	0	0	1	0	0	17	0	0	0	0	0
*A*8	0	0	0	0	0	0	0	4	0	0	1	0
*A*9	0	0	0	0	0	0	0	0	19	1	4	1
*A*10	0	0	0	0	0	0	0	0	1	14	0	2
*A*11	0	1	0	1	0	0	1	0	2	1	32	1
*A*12	0	0	0	0	0	0	0	0	0	1	1	16

6. Case Study

In the non-deep-learning method, the random forest classification method (RF) and K-nearest neighbor (KNN) classification perform well in action classification recognition. Therefore, the CNN-LSTM model proposed is compared with the RF and KNN methods. First of all, input the HAPT data set into RF and KNN. Then, segment the original

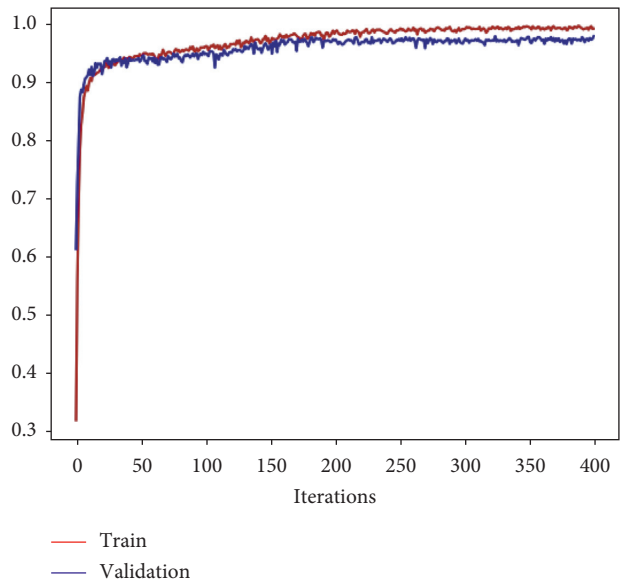

Figure 6: Accuracy curve of CNN-LSTM Model.

Table 9: The recognition accuracy, recall rate, and F value of various actions.

ID	Accuracy (%)	Recall (%)	F-measure (%)
$A1$	99.03	98.32	98.68
$A2$	97.78	98.73	98.35
$A3$	98.86	98.02	98.44
$A4$	90.76	91.85	91.30
$A5$	93.09	93.09	93.09
$A6$	99.78	99.56	99.56
$A7$	94.44	89.47	91.89
$A8$	100	100	100
$A9$	76.00	82.61	79.17
$A10$	82.35	77.78	80.00
$A11$	82.05	86.49	84.21
$A12$	88.89	80.00	84.21

Table 10: Average accuracy of various actions in CNN-LSTM, RF, and KNN models.

ID	RF (%)	KNN (%)	CNN-LSTM (%)
$A1$	99.90	88.10	99.03
$A2$	92.50	97.80	97.78
$A3$	90.20	99.40	98.86
$A4$	91.90	83.80	90.76
$A5$	90.80	87.50	93.09
$A6$	97.10	100	99.78
$A7$	71.30	66.70	94.44
$A8$	72.00	68.00	100
$A9$	51.30	38.60	76.00
$A10$	74.90	36.30	82.35
$A11$	59.20	33.70	82.05
$A12$	61.10	57.90	88.89

sensor data and calculate the mean value, variance, covariance, and 15 features. Finally, classify the basic actions and transition actions according to the clustering results. The classification results are shown in Table 10. It can be seen that the recognition rate of CNN-LSTM model is higher than that of RF and KNN methods for both basic actions and transition actions.

In addition to the comparison with RF and KNN classifier, our proposed model is also compared with a single CNN, a single LSTM, CNN-GRU, and CNN-BLSTM deep

Table 11: Average accuracy of different activities with five deep learning models.

ID	CNN (%)	LSTM (%)	CNN-BLSTM (%)	CNN-GRU (%)	CNN-LSTM (%)
*A*1	97.50	97.70	97.41	99.75	99.03
*A*2	97.25	97.10	95.65	98.99	97.78
*A*3	95.60	97.15	100	96.57	98.86
*A*4	91.26	90.26	91.96	81.99	90.76
*A*5	90.80	90.80	84.74	92.48	93.09
*A*6	99.67	98.58	100	99.78	99.78
*A*7	76.47	64.86	44.44	77.78	94.44
*A*8	100	66.67	66.67	50.00	100
*A*9	63.83	69.39	62.07	48.00	76.00
*A*10	84.85	70.27	80.00	52.94	82.35
*A*11	72.50	69.33	65.00	71.79	82.05
*A*12	83.30	70.27	70.59	55.56	88.89

Table 12: Average accuracy of the five models in this paper.

Method	Average recognition rate (%)
CNN	94.29
LSTM	93.22
CNN-BLSTM	92.73
CNN-GRU	93.34
CNN-LSTM	95.87

Table 13: Average accuracy of different methods on test set in the paper [35, 36].

Method	Average recognition rate
BLSTM [35]	87.5
DBN [36]	89.6
CNN-LSTM	95.8

learning models. Table 11 shows the average accuracy of various actions in five different depth models. As can be seen from Table 11, CNN-LSTM not only has a slightly higher recognition of basic movements than the other five models, but also has a significantly better recognition of transition movements, especially standing to sitting, sitting to lying, and standing to lying. Table 12 shows the recognition rates of different models on the test set. It can be seen from the table that the average recognition rate of the three models is higher than 90%, but the recognition effect of CNN-LSTM model is slightly better than that of CNN, LSTM, CNN-GRU, and CNN-BLSTM.

To prove the effectiveness of the CNN-LSTM deep learning model, it is also compared with other deep learning methods using the same dataset. Kuang [35] applied BLSTM to construct the behavior recognition model. Hassan et al. [36] used deep belief network (DBN) for human behavior recognition. We compared the performance with the approaches in [35, 36], with the result shown in Table 13. It follows that the proposed CNN-LSTM can achieve highest average recognition rate.

7. Conclusion

This paper explored the recognition method based on deep learning and designed the behavior recognition model based on CNN-LSTM. CNN learns local features from the original sensor data, and LSTM extracts time-dependent relationships from local features and realizes the fusion of local features and global features, fine description of basic and transition movements, and accurate identification of the two motion patterns.

The actions identified in this paper only include common basic actions and individual transition actions. In the next step, more kinds of actions can be collected and more complex actions can be added, such as eating and driving. And the individual recognition can be realized by considering the behavior differences of different users. Meanwhile, the deep learning model still needs to be optimized and improved. Studies show that the combination of depth model and shallow model can achieve better performance. Deep learning model has strong learning ability, while shallow learning model has higher learning efficiency. The collaboration between the two can achieve more accurate and lightweight recognition.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors would like to thank the support of the laboratory, university, and government. This research was funded by the National Key Research and Development Plan (No. 2017YFB1402103), the National Natural Science Foundation of China (No. 61971347), Scientific Research Program of Shaanxi Province (2018HJCG-05), and Project of Xi'an Science and Technology Planning Foundation (201805037YD15CG214).

References

[1] I. H. Lopez-Nava and M. M. Angelica, "Wearable inertial sensors for human motion analysis: a review," *IEEE Sensors Journal.*vol. 16, no. 15, 2016.

[2] Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum, "From action to activity: sensor-based activity recognition," *Neurocomputing*, vol. 181, pp. 108–115, 2016.

[3] T. Liu, F. Bingfei, and L. Qingguo, "The invention relates to a wearable motion sensor and a method for resisting magnetic field interference," 2017.

[4] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[5] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors (Switzerland)*, vol. 16, p. 1, 2016.

[6] X. Du, R. Vasudevan, and M. Johnson-Roberson, "Bio-LSTM: a biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1501–1508, 2019.

[7] Y. Huang, C. Wan, and H. Feng, "Multi-feature fusion human behavior recognition algorithm based on convolutional neural network and long short term memory neural network," *Laser Optoelectron. Prog*, vol. 56, p. 7, 2019.

[8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Berlin, Germany, 2010.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[11] C. Liu, J. Liu, Z. He, Y. Zhai, Q. Hu, and Y. Huang, "Convolutional neural random fields for action recognition," *Pattern Recognition*, vol. 59, pp. 213–224, 2016.

[12] K. Cho, M. Bart van, G. Caglar et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, Doha, Qatar, October 2014.

[13] M. Zeng, T. N. Le, Y. Bo et al., "Convolutional Neural Networks for human activity recognition using mobile sensors," in *Proceedings Of the 2014 6th International Conference On Mobile Computing, Applications And Services*, pp. 197–205, Austin, TX, USA, November 2015.

[14] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proceedings Of the 2015 ACM Multimedia Conference MM 2015*, pp. 1307–1310, Brisbane, Australia, October 2015.

[15] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *Proceedings of the 2015 IEEE International Conference On Systems, Man, and Cybernetics, SMC 2015*, pp. 1488–1492, Hong Kong, China, October 2016.

[16] 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.

[17] A. Murad and J. Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors (Switzerland)*, vol. 17, p. 11, 2017.

[18] J. Zhou, J. Sun, P. Cong et al., "Security-critical energy-aware task scheduling for heterogeneous real-time MPSoCs in IoT," *IEEE Transactions On Services Computing (TSC)*, vol. 12, p. 99, 2019.

[19] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.

[20] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Generation Computer Systems*, vol. 88, pp. 636–643, 2018.

[21] A. Ignatov, "Real-time human activity recognition from accelerometer data using Convolutional Neural Networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[22] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges," *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.

[23] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: a survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[24] S. Wu, G. Li, L. Deng et al., "$L1$-norm batch normalization for efficient training of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2043–2051, 2019.

[25] B. Almaslukh, J. Al Muhtadi, and A. M. Artoli, "A robust convolutional neural network for online smartphone-based human activity recognition," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 2, pp. 1609–1620, 2018.

[26] R. Yao, G. Lin, Q. Shi, and D. C. Ranasinghe, "Efficient dense labelling of human activity sequences from wearables using fully convolutional networks," *Pattern Recognition*, vol. 78, pp. 252–266, 2018.

[27] T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, and B. M. Eskofier, "Activity recognition in beach volleyball using a deep convolutional neural network," *Data Mining and Knowledge Discovery*, vol. 31, no. 6, pp. 1678–1705, 2017.

[28] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd international Conference on machine learning, ICML 2015*, vol. 3, pp. 2332–2340, Lille, France, July 2015.

[29] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, and Q. Sun, "Distributed consensus algorithm for events detection in cyber-physical systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2299–2308, 2019.

[30] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, *Improving Neural Networks by Preventing Co-adaptation of Feature Detectors*, arXiv preparation, Geneva, Switzerland, 2012.

[31] B. M. h. Abidine, L. Fergani, B. Fergani, and M. Oussalah, "The joint use of sequence features combination and modified weighted SVM for improving daily activity recognition," *Pattern Analysis and Applications*, vol. 21, no. 1, pp. 119–138, 2018.

[32] G. M. Weiss, J. W. Lockhart, T. T. Pulickal et al., "A smartphone-based activity recognition system for improving health and well-being," in *Proceedings of the 3rd IEEE International Conference On Data Science And Advanced Analytics, DSAA 2016*, pp. 682–688, Montreal, QC, Canada, October 2016.

[33] L. Wang and R. Liu, "Human activity recognition based on wearable sensor using hierarchical deep LSTM networks," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 837–856, 2019.

[34] W. Lu, F. Fan, J. Chu, P. Jing, and S. Yuting, "Wearable computing for internet of things: a discriminant approach for human activity recognition," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2749–2759, 2019.

[35] X. Kuang, *Human Behavior Recognition Based on Deep Learning and Wearable Sensor*, Nanjing University of Information Engineering, Nanjing, China, 2018.

[36] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, 2018.