HDL: Hierarchical Deep Learning Model based Human Activity Recognition using Smartphone Sensors

Tongtong Su* Huazhi Sun*, Chunmei Ma*, Lifen Jiang*, Tongtong Xu *
*School of Computer and Information Engineering, Tianjin Normal University
Email: mcmxhd@163.com

***Abstract*—With the development and popularization of smartphones, human activity recognition methods based on contact perception are proposed. The smartphones which are embedded with various sensors can be used as a platform of mobile sensing for human activity recognition. In this paper, we propose an automated human activity recognition network HDL with smartphone motion sensor units. The HDL network combines DBLSTM (Deep Bidirectional Long Short-Term Memory) model and CNN (Convolutional neural network) model. The DBLSTM model is first used to model long sequence data and ultimately generate a bidirectional output vector in a abstract way. The DBLSTM model is good at dealing with serialization tasks but poor in the ability to extract features. Hence, the CNN model is then used to extract features from the abstract vector. Finally, the output layer employs a softmax function to classify human activities. We conduct experiments on the Public domain UCI dataset. The experimental results show that the proposed HDL network achieves reliable results with accuracy and F1 score as high as 97.95% and 97.27%. Compared with other networks based on the same smartphone dataset, the accuracy of HDL is higher than S-LSTM and Dropout CNN network by 2.14% and 6.97% respectively.**

***Index Terms*—Human activity recognition, SmartPhones, Sensors, Deep Bidirectional Long Short-Term Memory, Convolutional neural network**

I. Introduction

With the rapid development of smartphones, which are built into various sensors, activity recognition based smartphone sensors has attracted a lot of research attention [1] [2] [3]. In [4], the authors present an accelerometer sensor-based approach for human activity recognition. This method uses a hierarchical scheme, where the recognition of ten activity classes is divided into five distinct classification problems. In [5], the authors propose an activity recognition system on a mobilephone, in which the uncertain time-series acceleration signal is analyzed by using hierarchical hidden Markov models. These methods obtain a comprehensive characteristics of sensory data, however, they cannot denote the internal relations of sequence data. Besides, these methods need adjusting the sensing device coordinates to reflect the motion characteristics of the object, which could lead to the neglect of some key features.

With the superior performance of deep learning in image detection [6] [7] [8] and speech recognition [9] [10], deep learning methods based on sequence data for activity recognition have been proposed. In [11], the authors assemble signal sequences of accelerometers and gyroscopes into a novel activity image. Then, Deep Convolutional Neural Networks (DCNN) is used to automatically learn the optimal features from activity images for activity recognition. In [12], the authors show the potential of CNNs and investigate basic parameters (such as max pooling, weight decay, or dropout). They report an accuracy of up to 88.19% on the Skoda dataset which is 4.41% better compared to the best conventional machine learning approach. In [13], a deep convolutional neural network (convnet) is proposed to perform efficient and effective HAR using smartphone sensors. Experiments show that convnet can achieve an overall performance of 94.79% on the test set with raw sensor data, and 95.75% with additional information of temporal fast Fourier transform of the HAR data set. The main disadvantage of above methods is that they cannot capture deep and fine-grained features, which lead to the poor performance of feature extraction. Besides, activity recognition using smartphones is a classic multivariate time sequences classification problem. Some typical models that are suitable for time sequence analysis, such as RNN,LSTM and BLSTM etc, should be selected. Therefore, these methods are not ideal in dealing with sequence data.

In this paper, we propose a multi-model fusion network HDL to perform human activity recognition, which has strong ability for modeling sequence data and extracting features. Firstly, the DBLSTM model that consists of multiple layers of BLSTM is used to capture the long-term sequence dependencies based on past and future contexts. Each BLSTM layer contains a forward LSTM and a backward LSTM, and the output of current BLSTM layer is transmitted to the next BLSTM layer. Each BLSTM layer which has an feature fusion can obtain more context information. As the BLSTM layer increases, the bidirectional output can be more deeply integrated. Therefore, we can get fine-grained feature representations at the last level. In addition, we train a single BLSTM layer on the same dataset and observe that deep networks give higher accuracy in very less number of training steps. As we can observe from Figure 7 that single layer network needs more epochs to converge, proving the advantages of depth of a

978-1-7281-2009-6/$31.00 paper N-19656.pdf

network. Finally, CNN is adopted to extract features from the fine-grained features. We evaluate the performance of our HDL network on an open public dataset and get the best results, with accuracy and F1 score as high as 97.95% and 97.27% respectively.

The followings are some of the key contributions and findings of our work:

1) We propose a novel human activity recognition network HDL, which combines DBLSTM (Deep Bidirectional Long Short-Term Memory) model and CNN (Convolutional neural network) model. It helps to improve the accuracy of sequence data classification.
2) We increase the number of layers of BLSTM to better capture the information of sequence data, which is helpful for extracting fine-grained features.
3) We introduce CNN into the DBLSTM model to extract features, which achieves remarkable results and also outperforms a wide range of baseline models.
4) We evaluate our proposed network with real data of multiple features, which are collected by sensors embeded into smartphones.

The rest of the paper is organized as follows: In Section II, we give a brief overview of the related work. In Section III, we present details of the proposed application. In Section IV, we explain the experimental setup and present our results. Section V draws the conclusions.

II. Related Works

Motivated by the challenges of recognizing human activities, a number of researchers have made many contributions. Up to now, these works are mainly divided into two categories which are vision-based and sensor-based approaches.

Vision based approaches The purely vision-based approaches mainly focused on the frames of videos to recognize human activities. In [14], the authors presented a new framework for human activity recognition from video sequences captured by a depth camera. They clustered hypersurface normals in a depth sequence to form the polynormal which was used to jointly characterize the local motion and shape information. In [15], the authors applied tracking techniques in a close environment to detect falls. They used a connected-components labeling to compute the silhouette of a person and extracted features such as the spatial orientation of the center of the silhouette or its geometric orientation. In [16], the authors proposed a Robust Non-Linear Knowledge Transfer Model (R-NKTM) for human action recognition. The R-NKTM was learned from dense trajectories of synthetic 3D human models and generalized to real videos of human actions. In [17], the authors stacked differences between projected depth maps as the depth motion maps where HOG was extracted as the global representation of a depth video. In [18], the authors developed a 3D CNN model for action recognition. Features from both spatial and temporal dimensions was constructed by performing 3D convolutions. These methods extracted meaningful features such as silhouettes or bounding boxes from the frames by means of computer vision techniques. This was of great significance to facilitate activity recognition.

Sensor based approaches Human activities are composed of the movements of human body which can be represented by sensory data collected from three-axis sensors e.g., accelerometer and gyroscope. Many researchers had investigated activity recognition with various wearable sensors [19] [20] [21] [22]. In [23], the authors used phone-based accelerometers and collected labeled data of 6 activities from 29 users. The classification accuracy ranged from 77.6% to 96.9%. In [24], the authors adopted a set of time-frequency domain features. The unsupervised strategies (e.g. mixture of Gaussian, DBSCAN etc.) could achieve around 90% accuracy when the number of activities is unknown. However, these physical models are dependent on domain expertise as the higher accuracy can only be achieved by extracting the correct features. Subsequently, some researchers proposed to leverage deep learning methods to recognize human activities. In [25], the authors proposed a new human activity recognition approach based on synergistic LSTM neural network. Experiment shows that the proposed approach was the best with recognition accuracy by 95.81%, higher than 91.53% of CNN and 90.47% of LSTM. In [26], the authors proposed a systematic feature learning method for HAR problem. This method adopted a deep convolutional neural networks (CNN) to automate feature learning from the raw inputs in a systematic way. In [27], a deep convolutional neural network (convnet) is proposed to perform efficient and effective HAR using smartphone sensors by exploiting the inherent characteristics of activities and 1D time-series signals. In [28], the authors proposed a generic deep framework (DeepConvLSTM) for activity recognition based on convolutional and LSTM recurrent units, which was suitable for multimodal wearable sensors. Compared with the DeepConvLSTM network, we propose a new network HDL based on hierarchical deep learning model. The HDL network uses deep BLSTM to get more fine-grained and distinguishing features.

III. The HDL Network

With the popularity of smart mobile devices, including phones and wearables, activity recognition based mobile perception has received increasing attention. The emphasis of human activity recognition is shifting from computer vision-based methods to sensor-based methods. Compared to previous activity recognition methods based on images, the sensory data is usually sequential and heterogeneous. In order to extract typical features from sensory data, we use the DBLSTM model which consists of many BLSTM layer. BLSTM is a special kind of recurrent neural network (RNN), which is capable of processing serialized information via its recurrent structure. BLSTM feeds the raw data to the hidden layer on a frame-by-frame way. Hence, we can obtain the initial features in the first layer. In order to get the fine-grained features, we increase the BLSTM layer to obtain more abstract representation of sequence data. In addition, CNN has shown its ability to extract features from visual images. Therefore,

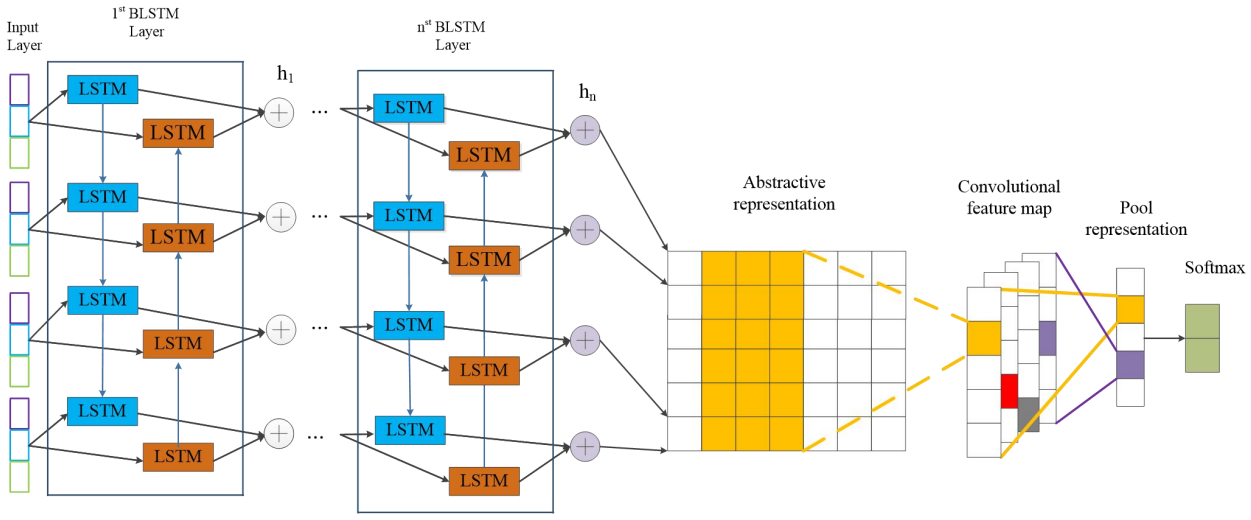

Fig. 1. The Architecture of HDL network. The whole architecture is divided into four parts. DBLSTM Layer is used for capture the information of original sensor data. CNN Layer is used to extract features from the output of last BLSTM Layer. Output layer that employs softmax function to classify human activities. Blocks of the same color in the convolutional feature map layer and the pool representation layer correspond to features of the same window.

deep layers of BLSTM and one layer of CNN are integrated to our new network: the HDL network. Figure 1 shows the architecture of the proposed network, in which the dot between first and second layer represents a concatentation. ie., the outputs of the forward and backward cells are concatenated together and fed to the backward and forward layers of the next upper layer. There is an explicit connection between backward and forward layers. When the output of DBLSTM layer is fed to the CNN layer, a max-over pooling layer is adopted to obtain a fixed length vector. After that, a fully connected layer and a softmax layer are performed on the obtained representation to classify human activities.

A. Input Layer

The definition of sequence data [29] contains two aspects. First, data is closely related to time and changes with time. Second, data is arranged in chronological order. Hence, the sensory data generated by smartphones is time labelled. For example, if we sample at the time t_0 with sampling window length of τ, the sensory data can be denoted as: $I = \{(\mathbf{x}_i^{'}, y^{(i)}, i = 1, 2, ..., N\}_{[t_0+(i-1)\tau]}$, where $\mathbf{x}_i^{'}$ denotes sample data that is sequenced according to the collecting order. Thus, $\mathbf{x}_i^{'}$ can be time labelled by "sequence tag" T_i. $y^{(i)}$ is the class label that $\mathbf{x}_i^{'}$ belongs to. $\mathbf{x}_i^{'}$ can be marked by the sequence tag T_i. The dimension of $\mathbf{x}_i^{'}$ is $K * L$, where K is the number of features in each frame, representing the heterogeneity feature of the sensory data. L indicates the number of frames in a sequence of $\mathbf{x}_i^{'}$. For human activity recognition, time labels should be considered as much as possible in the design of a algorithm, but existing methods usually cannot handle temporal information effectively. Therefore, this paper selects the BLSTM model as an unit for extracting the initial features of the time labelled sequence data.

B. Hierarchical Abstraction

After the sequence data is learned, for an input data I, the hidden layer can output a data matrix. The BLSTM model [30] [31] can effectively utilize the context information of sequence data through forward LSTM and backward LSTM. Therefore, BLSTM is an enhanced version of LSTM (Long-Short Term Memory) [32] [33]. LSTM designs the input gate i, the forget gate f and the output gate o to control how to overwrite the information by comparing the inner memory cell C when new information arrives. When information enters a LSTM network, we can judge whether it is useful according to relevant rules. Only the information that meets algorithm's authentication will be remained, and inconsistent information will be forgotten through Forget gate. Given an input sequence $x = (x_0, ..., x_t)$ at time t and the hidden states of a BLSTM layer $h = (h_0, ..., h_t)$ can be derived as follows.

The forget gate will take the output of hidden layer h_{t-1} at the previous moment and the input x_t at the current moment as input to selectively forget the information in the cell state C_t, which can be expressed as:

$$f_t = sigmoid(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{1}$$

The input gate cooperates with a tanh function together to control the addition of new information. tanh function generates a new candidate vector $\widetilde{C}_t$. The input gate generates a value in $[0, 1]$ for each item in $\widetilde{C}_t$ to control how much new information will be added, which can be expressed as:

$$C_t = sigmoid(f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t) \tag{2}$$

Where

$$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + b_t) \tag{3}$$

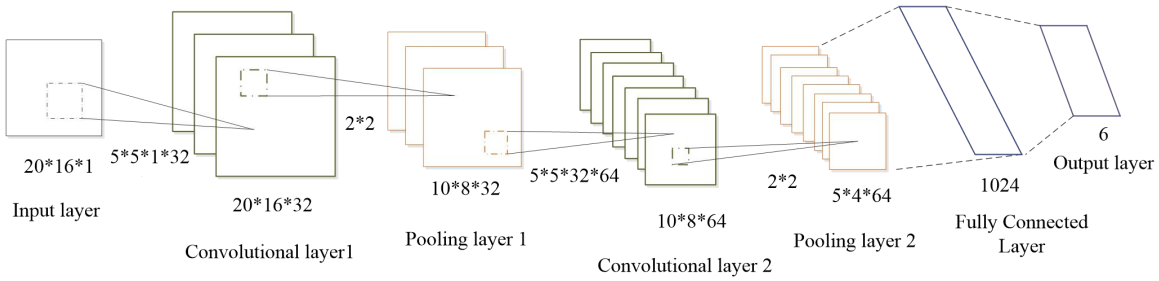

Fig. 2. The Structure of CNN for feature extraction. The structure presents the changes of data during the CNN process.

$$\widetilde{C}_t = \tanh(W_c x_t + W_c h_{t-1} + b_c) \tag{4}$$

The output gate is used to control how much of the current unit state will be filtered out, which can be expressed as:

$$o_t = sigmoid(W_{xo} x_t + W_{ho} h_{t-1} + b_o) \tag{5}$$

The output of the last LSTM unit is h_t, which can be expressed as:

$$h_t = o_t \tanh(c_t) \tag{6}$$

To achieve a hierarchical abstraction for representing an activity, we propose to use a deep neural network (formed by multiple stacked BLSTM). The DBLSTM structure is illustrated in Figure 1. The DBLSTM layers is used as hidden layers stacked between the input layer and output layer. The output vector of previous BLSTM is passed to the next BLSTM. Thus, by training 2^{st} BLSTM that takes all data matrixes h_{1t} outputted from 1^{st} BLSTM as input at time t, a second-level abstraction h_{2t} can be obtained. Accordingly, for input data I, the abstraction is defined as $X_{(i)} = h_{it}$, where i = 1, 2,$\cdots$ indicates the level. More BLSTM layers are added to the structure, more abstract features can be extracted. Consequently, the output from the last BLSTM layer can get n layers fusion, which allows us to understand human activities comprehensively.

For i^{st} BLSTM layer at time t, the output after passing through forward LSTM layer is represented as $\overrightarrow{h}_{it}$, given the input of sequence data x_t. Similarly, the output of backward LSTM is $\overleftarrow{h}_{it}$, which can be expressed as:

$$\overrightarrow{h}_{it} = \tanh(W_{x_i \overrightarrow{h}_i} x_t + W_{\overrightarrow{h}_i \overrightarrow{h}_i} \overrightarrow{h}_{it-1} + b_{\overrightarrow{h}_i}) \tag{7}$$

$$\overleftarrow{h}_{it} = \tanh(W_{x_i \overleftarrow{h}_i} x_t + W_{\overleftarrow{h}_i \overleftarrow{h}_i} \overleftarrow{h}_{it-1} + b_{\overleftarrow{h}_i}) \tag{8}$$

Finally, the output of the BLSTM h_{it} can be defined as the concatenation of $\overrightarrow{h}_{it}$ and $\overleftarrow{h}_{it}$, which can be expressed as:

$$h_{it} = \overleftarrow{h}_{it} + \overrightarrow{h}_{it} \tag{9}$$

wherein, dot $'\cdot'$ means the pointwise product. All the matrices W are the connection weights between two units. b is a bias vector.

C. Feature Classification

After modeling the sequence data by DBLSTM, we can obtain a sequence output. In accordance with [34], this sequence is converted into an activity image with size of $20 \times 16 \times 1$. Hence, the CNN model can be used to extract features from these images. The CNN model [35] [36] [37] consists of basic components such as input layer, convolutional layer, pooling layer, fully connected layer and output layer. The CNN model used in this paper is shown in Figure 2. The size of the first convolution kernels layer is (5, 5, 1). The size of the second convolution kernel layer of is (5, 5, 32). In order to ensure that the size after convolution and input size are the same, we use the padding operation. The pooling kernel used by the two pooling layers is (2, 2). There are 1024 hidden nodes in the fully connected layer, and the softmax function is used to classify human activities.

The most critical of the CNN model is convolutional layer and pooling layer, in which the convolutional calculation process is as shown in equation (10).

$$x_{i,k}^{l,j} = f(b_j + \sum_{a-1}^{m} w_{a,k}^{j} x_{i+(k-1)\times s+a-1}^{l-1,j}) \tag{10}$$

where $x_{i,k}^{l,j}$ is one of the ith unit of j feature map of the kth section in the lth layer, and s is the range of section. f represents a nonlinear function, here the $sigmoid$ function is used.

The pooling layer is a process that further reduces the dimensions of the matrix without destroying the intrinsic link of the data. Since average pooling adopts the average value in each convolution, max pooling is selected to extract the most significant elements in each convolution and then they are turned into feature vectors. Hence, the max pooling is used for dimensionality reduction in this paper, and the calculation process is as shown in equation (11).

$$x_i^{l,j} = max_{k=1}^{r}(x_{(i-1)\times s+k}^{l-1,j}) \tag{11}$$

D. Training

Training the proposed network contains a forward pass and a backward pass.

Forward Propagation The network is composed of deep BLSTM layer and one CNN layer, each of which presents

different structures and thus plays different role in the whole network. First, the forward propagation is conducted in i^{st} BLSTM layer at time t by equation (9). Then, we assemble the outputs of the last BLSTM layer into an image. After two convolution and pooling operations, a fully connection layer is followed for feature fusion and the fused results x_i^l are normalized by the softmax function to get each class probability $p(y)$:

$$x_i^l = \sum_j w_{j,i}^{l-1} sigmoid(x_i^{l-1}) + b_i^{l-1} \tag{12}$$

$$P(y=a) = \frac{e^{x^T wj}}{\sum_k e^{x^T wk}} \tag{13}$$

where a is a class label, x is a sample feature, y is label variable, and w is weight vector.

The objective function of our model is the cross-entropy based cost function. The goal of training the model is to minimize the cross entropy of the expected and actual outputs for all activities, which can be given as:

$$L = -\sum\sum y_i^k \ln a_i^k + (1-y_i^k)\ln(1-a_i^k) \tag{14}$$

Where i is the index of activity data. k is the number of class categories. The activities are classified into six categories in total.

Backward Propagation In this paper, we use the Back Propagation Through Time (BPTT) [12] algorithm to obtain the derivatives of the objective function with respect to all the weights, and minimize the objective function by stochastic gradient descent.

IV. Evaluation

In this section, we first determine the parameters of HDL network to obtain the optimal model through experiments which carry out an benchmark dataset. Then, we analyze the performance of the HDL network. Finally, to verify the effectiveness of the HDL network, we compare the performance of this network with some state-of-the-art methods.

A. Benchmark Dataset

In this paper, we use the Public domain UCI dataset [38]. The experimental data is collected by 30 users aged 19-48 wearing smartphones on the waist. The collected data is divided into 6 categories, including 3 static activities (standing, sitting, lying) and 3 dynamic activities (walking, going upstairs, going downstairs). With the accelerometer and gyroscope embedded into the phone, we get the three-axial linear acceleration together with three-axial angular velocity for twice. For the first time, the mobile phone was required to be placed on the left belt, but for the second time the mobile phone location was not required. The frequency of sampling is set to 50 Hz. There are totally 10929 samples which are then separated into small fragments by a sliding window set as 128 frames.

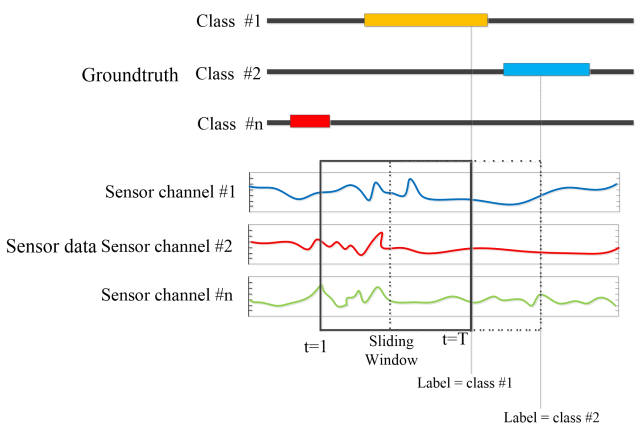

Fig. 3. Sequence labelling after segmenting the Opportunity dataset with a overlapping sliding window. The current data fragment overlaps the previous data fragment with 50%. The class label of the data fragment is set as the label of the last frame in each slide window.

Since the Public domain UCI dataset is recorded continuously, it is necessary to use a sliding window of fixed length T to segment the data. In order to add some redundancy during training and testing, the sliding window is overlapped by 50%. Repeating the operation above can generate a dataset suitable for the training and testing. In this paper, the length of the window is 128 ms, with a step size of 64 ms. In addition, the class label of the data fragment is set as the label of the last frame in each slide window, as shown in Figure 3. Since the sensory data are acquired from a set of sensors embedded into mobile devices. The dataset is multichannel time series. Therefore, after a slide window process, the shape of the dataset $SequenceNum \times channels$ can be converted into $sampleNum \times windowSize \times channels$.

B. Experimental Settings

In this paper, we design and implement a deep learning model for efficient classification of human activities. We use the *TensorFlow* framework and integrate the *Pycharm* development environment for experiments. In addition, the data samples are divided into two parts: one part is used to build a classifier, that is called the training dataset. The other is used to evaluate the classifier, that is called the testing dataset. For the Public domain UCI dataset, 7982 of the data are selected for training and the remaining 2947 for testing.

C. Parameter Optimization

In the deep learning system, the process of adjusting model parameters is manual and is mainly governed by experiences. In our experiments, to obtain the optimal model, we adjust the corresponding model parameters (including the number of neurons on the hidden layer, learning rate η, etc.), and explore different configurations to these parameters. These parameters settings in the experiment are shown in Table 1, and some important parameters are explained.

TABLE I
HYPERPARAMETERS OF THE HDL NETWORK

hyperparameter	values	hyperparameter	values
hidden units	32	CONV_KSIZE	5*5
learning rate(η)	0.0025	CONV_STRIDE	[1,1,1,1]
training epoches	150	CONV_NUM	32
hidden layer	3	POOL_KSIZE	2*2
l2 regularization(λ)	0.0015	POOL_STRIDE	[1,2,2,1]
dropout	0.75	POOL_NUM	64

After lots of experiments, three BLSTM layer and one CNN layer are adopted when building the HDL network for activity recognition task. For the HDL network: i) When η=0.0025, the accuracy can get a more higher value and stable state. Consequently, η=0.0025 is selected as the value of the learning rates for the HDL network. ii) As the number of hidden layer units increases, the loss cost gradually decreases. After H=32, the loss cost of the HDL network is stable. Consequently, H=32 is selected as the number of hidden layer units for the HDL network.

D. Performance Analysis of HDL

The confusion matrices on the Public domain UCI dataset for the HDL network is illustrated in Figure 4. The color from white to black represents the increasing percentage. It can be seen that the LAYING_DOWN and WALKING class are recognized best, while WALKING_UPSTAIRS, WALKING_DOWNSTAIRS are easily misrecognized as each other because triaxial acceleration and triaxial angular velocity producing by these two activities are quite similar from the values in others. In fact, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS are seemingly almost the same from the point of view of a device placed not required, which is how the dataset is gathered.

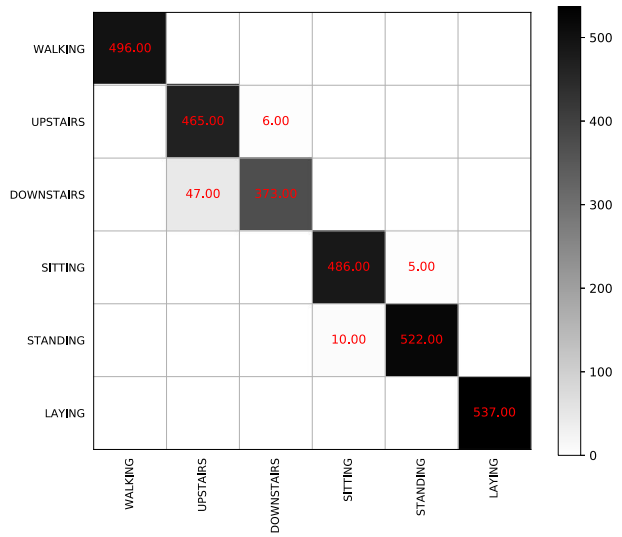

Fig. 4. Matrix confusion on testing dataset using the HDL network.

TABLE II
MODEL F1 SCORE YIELDED BY THE HDL NETWORK ON THE PUBLIC DOMAIN UCI DATASET.

	WK	WU	WD	ST	SA	LA
Precision	100%	91.9%	98.4%	97.6%	98.1%	100%
Recall	100%	98.3%	88.8%	99.0%	95.7%	100%
F1 score	100%	95.0%	93.4%	98.3%	96.9%	100%

The performances on the Public domain UCI dataset for the HDL network are illustrated in Table 2. The precision of six classes are in the range of 91.9% to 100%, and the recall of are in the range of 88.8% to 100%. The integral F1 score of the HDL network is 97.27%. This is in line with the performance evaluation indicators of Figure 4. The F1 score of WALKING and LAYING are up to 100%, and the F1 score of WALKING_UPSTAIRS and WALKING_DOWNSTAIRS are only above 95.0% and 93.4% respectively.

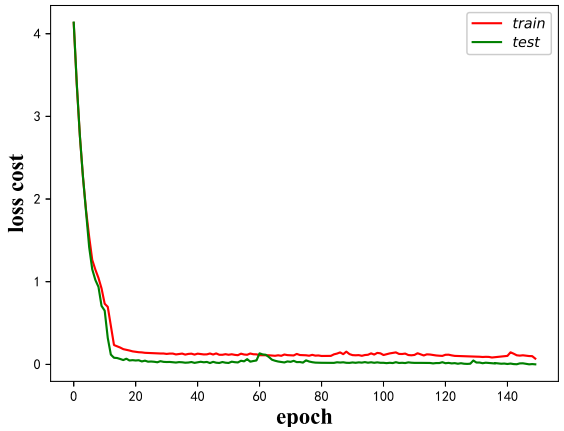

Fig. 5. The loss cost of the HDL network on the Public domain UCI dataset.

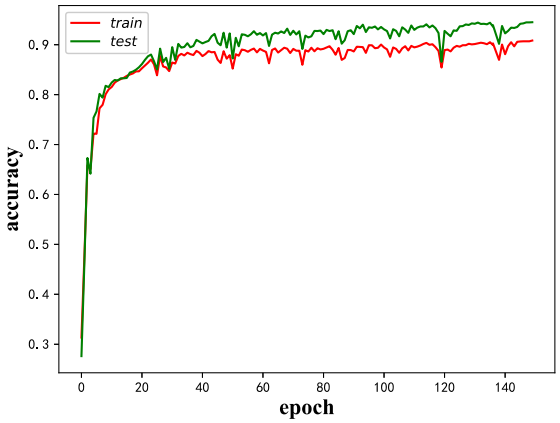

Fig. 6. The accuracy of the HDL network on the Public domain UCI dataset.

We evaluate the performance of the HDL network by the progress for running the network over multiple iterations. Firstly, we determine whether the phenomenon of overfitting

exists in our network by verifying the testing set. The experimental results in Figure 5 show that as the loss cost of training set is decreasing, so does the loss cost of the testing set. Thus, the HDL network fits the testing data well and we can conclude that there is no overfitting problem in our network. In addition, Figure 6 shows that the testing result is pretty good that achieves 97.95% accuracy. After epoch=25, the amplitude of the HDL network nearly remain steady.

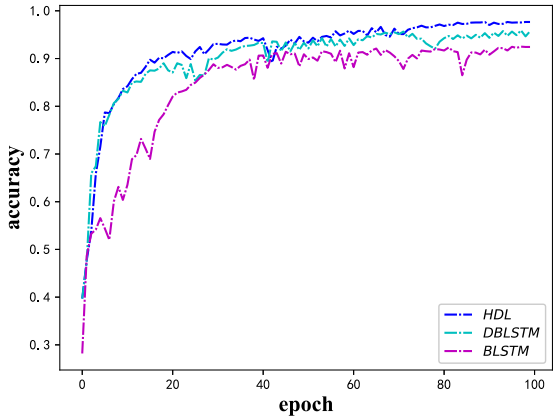

Fig. 7. The Comparison of Accuracy between single BLSM and DBLSM.

E. Comparison to the State of the Art

In order to objectively evaluate the accuracy and differentiation of the HDL network, we compare our network with two related works proposed by [25] and [34]. We choose them for comparison because they are not only the same activity classification tasks that used the same dataset ,but also both recent highly relative and representative works on mobile device, which achieve remarkable accuracy. The experimental results are shown in Table 3.

TABLE III
COMPARISON RESULTS ON THE PUBLIC DOMAIN UCI DATASET

method	accuracy
LSTM [25]	90.47%
CNN [25]	91.53%
S-LSTM [25]	95.81%
LSTM [34]	87.38%
BLSTM [34]	84.54%
CNN [34]	85.4%
Dropout CNN [34]	90.98%
BLSTM	95.70%
DBLSTM	96.75%
HDL	97.95%

From Table 3, we can observe that the HDL network performs better than other models in terms of accuracy, which can reach 97.95%. Firstly, the HDL network is more accurate than Dropout CNN because Dropout CNN is more suitable for processing image data. Additionally, Dropout CNN uses a fixed convolution kernel that cannot model longer sequence information, which is not conducive to the feature extraction of sequence data. Secondly, the HDL network is better than DBLSTM because HDL combines CNN to extract features on the contextual information obtained by DBLSTM. Finally, comparing deep BLSTM with single BLSTM, we observe that deep networks give higher accuracy in very less number of iterations, confirming the advantages of depth of a network. These results prove that the HDL network can offer a significant advantage across very different scenarios.

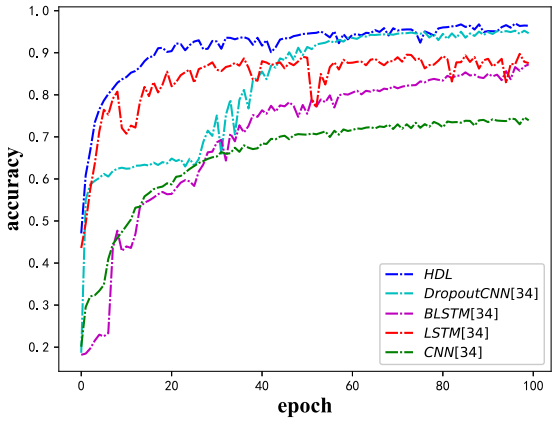

Fig. 8. The Comparison of Accuracy with different models.

As shown in Figure 8, the accuracy of each model shows an overall upward trend. The accuracy of HDL in the testing set increases fastest and remains stable after 25 iterations. The accuracy of Dropout CNN is slower than HDL at the beginning. As the number of iterations increase, the accuracy of Dropout CNN begin to increase. LSTM is unstable and its accuracy is lower than Dropout CNN and HDL. The accuracy of BLSTM keeps improving, but it needs more epochs to become stable. CNN increases slowly, and it has the worst accuracy on the testing set. Therefore, HDL can achieve a recognition accuracy of 97.95% for sequence data, which is sufficient for activity recognition.

V. CONCLUSION

In this paper, we mainly introduce a combined network called HDL that is made up of DBLSTM model and CNN model. The DBLSTM model is used to model long sequence data based on past and future contexts. Then, the CNN model is used to extract features from the abstract vector. In our experiments, the HDL network is able to improve the accuracy by 2.14% and 6.97% respectively for the public domain UCI dataset in comparison with previous work. Hence, we believe that the proposed HDL can be used as a powerful tool for human activity recognition problems.

As for future work, there are two aspects we can explore: 1) With the development of technology, more sensors can be

integrated on smartphones. We can improve the extension of sensor data to more axes for similar activities classification. ii) Transfer learning approach based on existing models to perform activity recognition on large-scale data may be a potential working direction.

VI. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NO.61702370, No.61602345), the Research project of Tianjin science and technology development strategy (NO.17ZLZXZF00530), the Natural Science Foundation of Tianjin (18JCYBJC85900, 18JCQNJC70200, NO.15/JCQNJC01400, NO.16JCQNJC01100), the Doctoral Fund of Tianjin Normal University (NO.043/135202XB1615, NO.043/135202XB1705) and the 131 three-level candidates of Tianjin Normal University (NO.043/135305QS20).

References

[1] 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.

[2] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[3] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones ," *Procedia Computer Science*, vol. 34, pp. 450–457, 2014.

[4] Y. Zheng, *Human Activity Recognition Based on the Hierarchical Feature Selection and Classification Framework*. Hindawi Publishing Corp., 2015.

[5] Y. S. Lee and S. B. Cho, *Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer*. Springer Berlin Heidelberg, 2011.

[6] S. P. Mohanty, D. P. Hughes, and M. Salath, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, 2016.

[7] T. Dobhal, V. Shitole, G. Thomas, and G. Navada, "Human activity recognition using binary motion image and deep learning ," *Procedia Computer Science*, vol. 58, pp. 178–185, 2015.

[8] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," pp. 5–10, 2016.

[9] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," vol. 38, no. 2003, pp. 6645–6649, 2013.

[10] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2016, pp. 167–174.

[11] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," pp. 1307–1310, 2015.

[12] M. Zeng, T. N. Le, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *International Conference on Mobile Computing, Applications and Services*, 2015, pp. 197–205.

[13] C. A. Ronao and S. B. Cho, *Human activity recognition with smartphone sensors using deep learning neural networks*, 2016, vol. 59.

[14] X. Yang and Y. L. Tian, "Super normal vector for activity recognition using depth sequences," in *Computer Vision and Pattern Recognition*, 2014, pp. 804–811.

[15] A. Mihailidis, B. Carmichael, and J. Boger, "The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home," *IEEE Transactions on Information Technology in Biomedicine A Publication of the IEEE Engineering in Medicine & Biology Society*, vol. 8, no. 3, p. 238, 2004.

[16] H. Rahmani, A. Mian, and M. Shah, "Learning a deep model for human action recognition from novel viewpoints," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2018.

[17] X. Yang, C. Zhang, and Y. L. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *ACM International Conference on Multimedia*, 2012, pp. 1057–1060.

[18] J. Shuiwang, Y. Ming, and Y. Kai, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[19] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.

[20] H. Gong, L. Yu, and Z. Xue, "Social contribution-based routing protocol for vehicular network with selfish nodes," *International Journal of Distributed Sensor Networks*, vol. 2014, no. 1, pp. 1–12, 2014.

[21] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Transactions on Knowledge & Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.

[22] M. N. Nyan, F. E. H. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection," *Journal of Biomechanics*, vol. 41, no. 16, pp. 3475–3481, 2008.

[23] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *Acm Sigkdd Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[24] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6067–6074, 2014.

[25] C. D. G. J. ZHU, Lianzhang and Z. Hongxia, "Research on human action recognition based on synergistic lstm neural network," *Computer Technology and Development*, 2018.

[26] B. Y. Jian, M. N. Nguyen, P. P. San, L. L. Xiao, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *International Conference on Artificial Intelligence*, 07 2015.

[27] C. Ann Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, 04 2016.

[28] F. J. Ordonez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[29] L. Qi and S. Wu, "Review on time-series data visualization," *Microcomputer & Its Applications*, 2015.

[30] M. Wollmer, F. Eyben, B. Schuller, and G. Rigoll, "A multi-stream asr framework for blstm modeling of conversational speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 4860–4863.

[31] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlter, and H. Ney, "A comprehensive study of deep bidirectional lstm rnns for acoustic modeling in speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[32] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[33] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1942–1950.

[34] X. h. Kuang, J. He, Z. Hu, and Y. Zhou, "Comparison of deep feature learning methods for human activity recognition," *Application Research of Computers*, vol. 35, no. 9, p. 7, 2018.

[35] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, no. 2, pp. 119–130, 1988.

[36] R. Z. Liang, G. Liang, W. Li, Q. Li, and J. Y. Wang, "Learning convolutional neural network to maximize pos@top performance measure," 2017.

[37] Hong, SHAO, Shuang, CHEN, Jie-yi, ZHAO, Wen-cheng, and Tian-shu, "Face recognition based on subset selection via metric learning on manifold," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 12, pp. 1046–1058, 2015.

[38] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," 01 2013.