

Sequential Weakly Labeled Multiactivity Localization and Recognition on Wearable Sensors Using Recurrent Attention Networks

Kun Wang¹, Jun He¹, *Member, IEEE*, and Lei Zhang¹

Abstract—With the popularity and development of the wearable devices such as smartphones, human activity recognition (HAR) based on sensors has become as a key research area in human computer interaction and ubiquitous computing. The emergence of deep learning leads to a recent shift in the research of HAR, which requires massive strictly labeled data in supervised learning scenario. In comparison with video data, activity data recorded from accelerometer or gyroscope are often more difficult to interpret and segment. Recently, several attention mechanisms are proposed to handle the weakly labeled human activity data, which do not require accurate data annotation. However, these attention-based models can only handle the weakly labeled dataset whose sample includes one target activity, as a result it limits efficiency and practicality. In the article, we propose a recurrent attention networks (RAN) to handle sequential weakly labeled multiactivity recognition and location tasks. The model can repeatedly perform steps of attention on multiple activities of one sample and each step is corresponding to the current focused activity. The effectiveness of the RAN model is validated on a collected sequential weakly labeled multiactivity dataset and the other two public datasets. The experiment results show that our RAN model can simultaneously infer multiactivity types from the coarse-grained sequential weak labels and determine specific locations of every target activity with only knowledge of which types of activities contained in the long sequence. It will greatly reduce the burden of manual labeling.¹

Index Terms—Human activity recognition (HAR), recurrent attention networks, weakly labeled data, wearable sensors.

I. INTRODUCTION

HUMAN activity is unique, as the information inferred from raw activity data has been proved to be very critical in human activity recognition (HAR) [1], health support [2], and smart homes [3] to name a few. With the popularity and

development of the wearable devices such as smartphones, human activity can be captured using a variety of motion sensors such as accelerometer and gyroscope worn on various parts of the body, which provides convenient interface between humans and machines [4], [5]. HAR can perform automatic detection of various physical activities performed by people in their daily lives. The traditional machine learning approaches such as support vector machine and hidden Markov model, based on hand-crafted features [6]–[9], have been extensively used in the HAR field.

Due to the emergence of deep learning, there has been a recent shift in the use of machine learning techniques. Deep learning methods can learn the features automatically from the data which avoids the problem of the hand-crafted features in shallow learning field. Deep learning approaches such as convolutional neural networks (CNN) [10]–[13] and recurrent neural networks (RNN) [14]– have proven to be more effective than the shallow learning techniques in discovering, learning, and inferring complex activity from data. However, deep learning often requires a large amount of annotated or “ground truth labeled” training data, which is one main challenge for supervised HAR tasks. Actually, ground truth annotation is an expensive and tedious task, in which the annotator has to carefully skim through the raw sensor data and manually label all activity instances. In addition, the data recorded from inertial sensors such as accelerometer or gyroscope are often more difficult to interpret than data from other sensor, such as cameras.

Intuitively, it is much easier for an annotator to know whether the target activity occurs in a long sensor sequence. If we can infer activity types from the coarse-grained labels and determine specific locations of every labeled target activity with only knowledge of which types of activities are contained in the long sequence, it will greatly alleviate the burden of sensor data annotation. The dataset with coarse-grained labels is referred to as weakly labeled dataset, in which raw sensor data was roughly segmented. Each sample may include one or multiple target activities, as well as the background activity. We tackle the above challenges from a novel aspect, i.e., attention, which recently has been studied extensively in various research areas such as computer vision and natural language processing (NLP). In our previous work [17], an end-to-end-trainable attention module was embedded into CNN architecture to identify interesting activity from weakly labeled dataset for HAR. Without need of strict annotation, the attention-based CNN can greatly facilitate

Manuscript received August 25, 2020; revised February 1, 2021; accepted April 10, 2021. Date of publication June 22, 2021; date of current version July 19, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61203237 and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191371. This article was recommended by Associate Editor X. Xu. (*Corresponding author: Lei Zhang.*)

Kun Wang and Lei Zhang are with the School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, China (e-mail: iskenn7@gmail.com; leizhang@njnu.edu.cn).

Jun He is with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: jhe@nuist.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/THMS.2021.3086008>.

Digital Object Identifier 10.1109/THMS.2021.3086008

¹The code of our work is available at <https://github.com/KennCoder7/RAN>

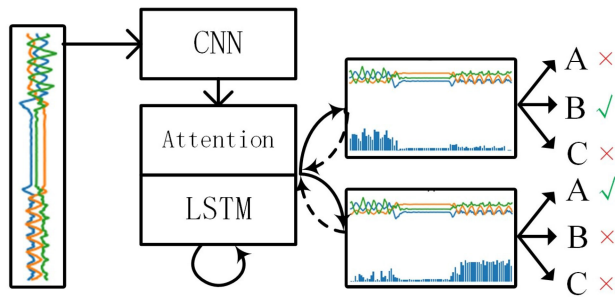


Fig. 1. Overall architecture of our model, which can produce an attention map of the focused activity at every step.

the process of sensor data collection. However, the attention-based CNN, in which the soft attention mechanism is used to enhance the target activity and suppress the irrelevant background activity, can only deal with the weakly labeled samples that include one target activity. The CNN that combines soft attention fails to capture the arranging order of multiple activity instances, as a result it limits efficiency and practicality. Therefore, the new challenge is that whether one can simultaneously recognize and locate multiple target activities from one sample of the weakly labeled HAR dataset.

To tackle this challenge, one feasible proposal is that enabling the mechanism of attention to become “recurrent.” Recently, Xu *et al.* [18] proposed a model to handle the task of automatically generating captions for an image, which can repeatedly perform steps of attention on multiple objects of an image and generate the weighted feature of the current focused object according to its previous observations.

In this article, edified by this notion [18], we propose a recurrent attention networks (RAN) to recognize and locate multiactivity types in sequential weakly labeled sensor samples. To the best of our knowledge, attention has seldom been combined with RNN in HAR scenario. This is the first article to leverage RAN to deal with sequential weakly labeled sensor data. The RAN model that consists of CNN, long short-term memory (LSTM) and attention module is shown in Fig. 1. For sequential activity recognition tasks, processing one sample usually consists of T steps. At each step t , the model needs to produce an attention map of the current activity and its corresponding attention feature. Then the trained feature must change accordingly to represent different activities, therefore the recurrent structure can be naturally exploited to provide the conditional information for the variable feature.

Our main contribution is threefold. First, we for the first time propose an efficient RAN model in HAR scenarios, which can simultaneously infer one or multiple activity kinds from the coarse-grained labels and determine specific locations of every target activity with only knowledge of what kinds of activities contained in the long sequence. Second, we perform extensive experiments to verify that the attention method achieves comparable performance with standard CNN, across the public benchmark UniMiB-SHAR and OPPORTUNITY datasets, as well as the sequential weakly labeled HAR dataset. Third, visualizing analysis of attention weights along temporal dimension

is provided to improve the comprehensibility of the sensor data annotation. The proposed method can effectively aid to the collection of “ground truth labeled” training data.

The remainder of this article is structured as follows. An overview of related works appears in Section II. In Section III, we describe the SWLM dataset collected for this research and two public datasets. Section IV propose our RAN model. The experimental results and discussion are then presented in Section V. Finally, Section VI concludes this article.

II. RELATED WORKS

HAR, has emerged as a key research area in human computer interaction (HCI) and ubiquitous computing. HAR can be seen as a typical pattern recognition problem, which has made tremendous progress by adopting shallow learning algorithms. In [6], Bao and Intille found that accelerometer sensor data are suitable for activities recognition. Four types of features (mean, energy, frequency, and domain entropy) were extracted manually from accelerometer data and activity recognition on these features was performed using decision table, instance-based learning (IBL or nearest neighbor), C4.5 decision tree, and Nave Bayes classifiers [19]. Kwapisz *et al.* [8] also used the accelerometer sensor of mobile devices to extract features, and six different hand-crafted features were generated and then fed into the classifiers such as decision trees (J48), multilayer perceptions (MLP), and logistic regression. However, the features of these shallow algorithms are usually extracted via a hand-crafted way, which heavily relies on domain knowledge or human experience and has low performance in distinguishing similar activities such as walking upstairs and walking downstairs [20]. Besides, choosing suitable features and extracting features from sensor data manually are both difficult and laborious.

The emergence of deep learning tends to overcome above drawbacks, and the features can be learned automatically through convolutional networks instead of being manually designed [21]. For instance, Chen and Xue [11] fed raw signal into a sophisticated CNN, which had an architecture composed of three convolutional layers and three max-pooling layers. Furthermore, Jiang and Yin [12] converted the raw sensor signal into 2-D signal image by utilizing a specific permutation technique and discrete cosine transformation, then fed the data into CNN to classify the signal image equaling to its desired activity recognition. Ordonez and Roggen [14] proposed the DeepConvLSTM model comprised of CNN and LSTM recurrent units, which outperforms CNN. Kumar *et al.* [15] proposed an approach to perform multimodal gait recognition by fusing motion sensor and video data. The signals of the motion sensors are modeled using an LSTM network and corresponding video recordings are processed using a 3-D CNN. Kim *et al.* [16] proposed the deepGesture model to perform arm gesture recognition based on motion sensors. They used a CNN with four convolutional layers to learn features in raw sensor data, and then the features extracted by the CNN are used as an input of gated recurrent unit (GRU) to capture long-term dependency and model motion sensor signals. Here, GRU can be seen as a lightweight LSTM structure. Teng *et al.* [22] proposed a layer-wise CNN with

local loss for the use of HAR task which shows that local loss works better than global loss for tested baseline architectures. A state-of-the-art survey [23] on HAR based on wearable sensors depicted details about the performance of current deep learning models. However, these methods that belong to supervised learning [24] requires massive data with perfect ground-truth to train models.

As the annotator has to skim through the raw sensor data and manually label all activity instances, ground truth annotation is an expensive and tedious task. In comparison with other sensors, such as cameras, activity data recorded from an accelerometer or gyroscope is also often more difficult to interpret and segment. Strictly labeling samples of sensor data needs much more manpower and computing resources. Recently, some semisupervised and weakly supervised learning approaches were proposed to improve the efficiency of the ground truth annotation tasks in HAR. Zeng *et al.* [25] presented the semisupervised methods based on CNN that can learn from both labeled and unlabeled data. Recent researches in computer vision, machine translation, speech recognition, and image caption have witnessed the success of attention mechanism [26], [27]. The attention idea can be exploited to handle weakly labeled HAR data, which does not require the strict data annotation. Inspired by the notion, He *et al.* [28] proposed a weakly supervised model based on recurrent attention learning, and this method can deal with weakly labeled HAR data by utilizing an agent to adaptively select from sequence of locations and then extract desired information. Besides, in the previous work [17], we proposed a soft attention CNN model via measuring the compatibility between local features and global features, which can amplify the salient activity information and suppress the irrelevant confusing information. Nevertheless, the above attention methods have the limitation that can only handle the weakly labeled samples whose segments include one labeled activity.

III. DATASET

A. UniMiB-SHAR Dataset

We utilize the public dataset to validate that our model can deal with traditional activity recognition tasks. The UniMiB-SHAR dataset consists of 17 daily activities aggregated from 30 volunteers. The data are recorded from a Samsung smartphone, which collects 3-axial linear acceleration at a constant rate of 50 Hz. We use the method mentioned in [29] to preprocess this dataset. We divided each dataset into training set and test set by distinguishing different subjects. Thirty volunteers' data are divided into two parts where 20 subjects are for training and 10 for test. A fixed length window of 151 is used to segment the data.

B. Sequential Weakly Labeled Multiactivity Dataset

We collect the sequential weakly labeled multiactivity (SWLM) dataset to perform the sequential weakly labeled HAR tasks. The dataset includes three types of activities: "chest stretch," "arm lateral stretch," and "arm vertical stretch," and we use "A," "B," and "C" to denote the three activities. The

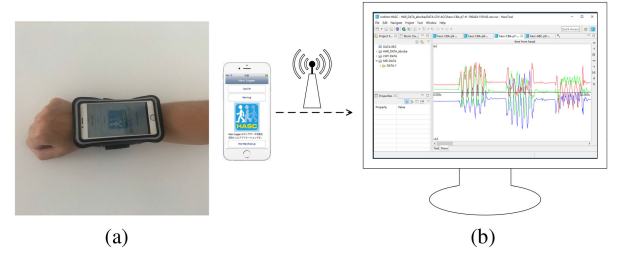


Fig. 2. (a) Smartphone is worn on the right wrist. (b) Uploading the data collected from smartphone to computer terminal by HASC Logger [30].

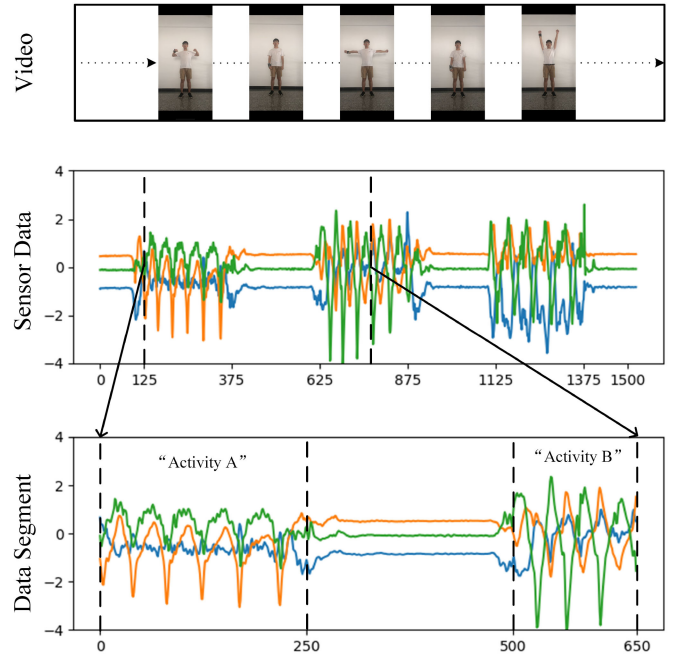


Fig. 3. Sample is labeled as "A-B." The activity between activity A and activity B is defined as "standing," which represents the background activity.

sensors data are collected from 3-axis accelerometer of iPhone tied to 10 subjects' right wrist as shown in Fig. 2(a). The subjects do the above three actions in the order "A-B-C" and "C-B-A." Each activity lasts five seconds (about five times), and there is a time gap between two types of activities. The smartphone has a sampling rate of 50 Hz. As can be seen in Fig. 2(b), the whole process of collection is supported by a mobile application named HASC Logger [30] which records the data from accelerometer and then uploads the data to computer terminal.

We divide raw data by distinguishing different subjects, and seven participants' data are used for training and the rest three subjects' data for test. Then we use a fixed length sliding window of 650 to segment the data. The whole process is illustrated in Fig. 3. Finally, the dataset consists of nine different types of samples: "A," "B," "C," "A-B," "B-C," "C-A," "C-B," "B-A," and "A-C." Here, each sample should correspond to its sequential weak labels. For example, the sample "A-B" is assigned with the sequential label ("start," "A," "B," "end").

C. OPPORTUNITY Dataset

To demonstrate the generality and superiority of the proposed method, we try to evaluate the performance on a large public HAR dataset namely OPPORTUNITY, which is devised to benchmark various HAR algorithms. The OPPORTUNITY dataset contains naturalistic human activities collected in a very rich sensor environment where each subject performed daily morning activities. In the article, we focus on data collected from on-body sensors consisting of inertial measurement units (IMU) and triaxial acceleration. The data are composed of the recordings of 4 subjects, and each subject was asked to perform 17 different daily activities with 20 repetitions. The first three subjects' data are used for training set and the rest for test set.

We preprocess the OPPORTUNITY dataset in both cases, in which sliding window techniques with two different window sizes are used. For the first case, i.e., the supervised case, we use a short length window of 64 to segment the data in order to build a normal HAR dataset with strict labels, in which each sample only contains one target activity without any background activity. For the second case, i.e., the weakly supervised case, a long length window of 600 is used to segment the sensor data, which yields a weakly labeled dataset. In this case, each sample may contain multiple activity instances, as well as the background activity. Compared with the SWLM dataset, this processed dataset reflects the concept of the weak label discussed in [17], in which one sample contains not only the target activity but also a large amount of background activities. More than that, one sample contains multiple types of activities that can be regarded as the combination between the target activity and the background activity. For example, we can segment a sequence and obtain a sample that contains the activity "open door" and "close door." In this case, the activity "close door" can be seen as background activity relative to the target activity "open door" when the model is trying to recognize "open door."

IV. MODEL

The aim of the proposed model is to recognize and locate multiactivity types in weakly labeled sensor data. The model consists of CNN, LSTM, and attention module. The CNN plays a role of feature extractor that acquires the feature vectors from sensor data. The combination of LSTM and attention module perform twofold functions including determining the location supposed to be paid attention to and generating classification result that matches the corresponding location. Utilizing above methods that mainly inspired by [18], we can visualize "where" and "what" the attention focus on.

A. CNN: Feature Extractor

CNN, which has great potential to identify the various salient patterns of HAR's signals, is used in order to extract features from the raw inputs. CNN maps the input to a set of feature vectors by convolutional kernels

$$\mathbf{a} = \{a_1, a_2, \dots, a_L\}, a_i \in R^D \quad (1)$$

where L denotes the numbers of feature vectors. Each of vector a_i is D -dimensional representation corresponding to the sensor data.

A classical CNN architecture consists of convolutional layers, pooling layers, and fully connected layer. In our model, the above feature vectors are not fed into the fully connected layer, because we do not need to output the classification probability at this stage. Here, the feature vectors are used as the input of attention module and LSTM. The CNN plays a role of an encoder. In the case of the sequential weakly labeled dataset, the shorthand description of the CNN feature extractor is: $Conv_1(16) - Pool_1 - Conv_2(32) - Pool_2 - Conv_3(64) - Pool_3 - Conv_4(128)$, where $Conv_s(k)$ denotes a convolutional layer s with k feature maps (i.e., the number of filters) and $Pool$ a max pooling layer. We apply 1-D convolution along temporal dimension and regard the sensor modalities as the channels, which can ensure the following attention mechanism can compute the weighted score along temporal feature vectors. Referring to the setting in the literature [13], [14], we set the kernels size to 5.

B. LSTM: Decoder

LSTM [31] is a special type of RNN. LSTM has the form of a chain of repeating modules of neural network, which can be utilized for multiactivity recognition. The implementation of LSTM is presented as

$$\begin{cases} \mathbf{i}_t = \sigma(W_i \mathbf{y}_{t-1} + U_i \mathbf{h}_{t-1} + Z_i \mathbf{z}_t + b_i) \\ \mathbf{f}_t = \sigma(W_f \mathbf{y}_{t-1} + U_f \mathbf{h}_{t-1} + Z_f \mathbf{z}_t + b_f) \\ \mathbf{c}_t = \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \tanh(W_c \mathbf{y}_{t-1} + U_c \mathbf{h}_{t-1} + Z_c \mathbf{z}_t + b_c) \\ \mathbf{o}_t = \sigma(W_o \mathbf{y}_{t-1} + U_o \mathbf{h}_{t-1} + Z_o \mathbf{z}_t + b_o) \\ \mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{c}_t) \end{cases} \quad (2)$$

where $\mathbf{i}_t, \mathbf{f}_t, \mathbf{c}_t, \mathbf{o}_t, \mathbf{h}_t$ denote the input, forget, memory, output, and hidden state of the LSTM, respectively. W, U, Z , and b are weight matrices and biases learned in the training phase.

In the previous HAR methods [14]–[16], the LSTM is used to encode the input data or the feature vectors generated from earlier subnetworks, and then feed the encoded vectors to the subsequent classifying networks in order to output classification results. In the article, the LSTM module plays a role of decoder that generates one classification results at each step t conditioned on the context vector \mathbf{z}_t , the preceding hidden state \mathbf{h}_{t-1} and the previously generated classification result \mathbf{y}_{t-1} .

The structure of the combination of LSTM and attention module is shown in Fig. 4. The LSTM enables the attention to become "recurrent," because the hidden state varies as the output RNN advances in its output sequence. That is to say, "where" the network looks next depends on the sequence of classification results that has already been generated. Repeating attention on the samples of weakly labeled sensor data containing multiple types of activities can implement the multiple activities recognition task. Besides, it can locate the target activities with sequential weak labels by weighing up the relative importance of different location of sensor data. At step t , the model produces an attention map of the current focused activity sample and its corresponding weighted feature vectors (i.e., the context vector \mathbf{z}_t) by the

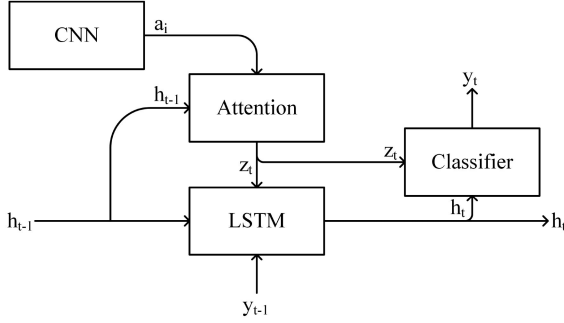


Fig. 4. Structure of the combination of LSTM and attention module.

attention mechanism. The context vector z_t , which is a dynamic representation of the relevant part of the sensor data input, is computed from the feature vectors a_i . Then the attention feature vectors replace the original feature vectors produced from the raw sensor data by CNN to participate in the loops of LSTM and then be fed into the classifier. Since the number of time-step should exceed that of the interesting activities contained in the longest sequential weakly labeled sample, it is set to 10 in our implementation. The number of LSTM cells is equal to the size of time-step and the memory units is 128. All cells use logistic sigmoid function as the gate activations, and hyperbolic tangent for the other activations. Each LSTM cell takes as inputs the connection of the previously generated classification result and the context vector, and the hidden state of the previous time. The initial memory state and hidden state of the LSTM are prebuilt by an average of the feature vectors fed through two different multilayer perceptron

$$c_0 = f_{init.c} \left(\frac{1}{L} \sum_i^L a_i \right), h_0 = f_{init.h} \left(\frac{1}{L} \sum_i^L a_i \right) \quad (3)$$

Equation (3) indicates that in the first step ($t=1$), the weight score is based on the feature vector a_0 totally.

At the end of each time step, a deep output layer (i.e., a dense layer) [32] f is used to compute the output probability, saying it is a classifier which cues from the context vector and the hidden state

$$p(y_t|a, y_{t-1}) \propto f(z_t + h_t). \quad (4)$$

C. Attention Module

In this section, we discuss the details of attention mechanism, that is to say how the context vectors are computed from the feature vectors $a_i, i = 1, \dots, L$ corresponding to the features extracted at different temporal locations of sensor data. The attention mechanism produces a positive weight α_{ti} which measures the relative importance of the feature vector a_i for each temporal location i . Closely following the one used in [18], the weight α_{ti} of each feature vector a_i is computed by using a multilayer perceptron conditioned on the previous hidden state h_{t-1}

$$e_{ti} = W_{att}(\hat{a}_i, (W_h h_{t-1} + b_h)) \quad (5)$$

where the \hat{a}_i is the projection of the feature vector a_i in order to have the same dimension with the hidden state h_{t-1} . W_{att} , W_h , and b_h are the learned weight matrices and biases. The e_{ti} is the weight score of temporal location i and the t is referred to time step. After the computing process, we have a set of score $S(a_i, h_{t-1}) = \{e_{t1}, e_{t2}, \dots, e_{tL}\}$, which are then normalized into $A_t = \{\alpha_{t1}, \alpha_{t2}, \dots, \alpha_{tL}\}$ by a softmax function

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^L \exp(e_{tj})}. \quad (6)$$

Then the normalized weights $\alpha_{ti}, i = 1, \dots, L$ are used to produce the context vector z_t by element-wise weighted averaging as proposed by [33]

$$z_t = \sum_{i=1}^L \alpha_{ti} \cdot a_i. \quad (7)$$

In essence, the methods computing the soft attention weighted feature vectors are based on a deterministic attention model, which discredits irrelevant activity information by multiplying the corresponding features map with a lower weight. Due to different weighted parameters, the noticeable attention part is enhanced while the less significant attention part is weakened. By this way, the weakly labeled data can be effectively recognized. Besides, the whole model is differentiable, which make an end-to-end training feasible by utilizing standard back-propagation.

D. Optimization

We utilize a doubly stochastic regularization [18] that encourages the model to pay attention equally to different parts of the sensor data. At time t , the attention at every point sums to 1 (i.e., $\sum_i \alpha_{ti} = 1$), which potentially result in ignoring some parts of the inputs by decoder. In order to alleviate this, we encourage $\sum_t \alpha_{ti} \approx \tau$ where $\tau \geq \frac{L}{D}$. So the final loss function is defined as

$$L_d = -\log(p(y|a)) + \sum_i^L \left(\tau - \sum_t^c \alpha_{ti} \right)^2 \quad (8)$$

where the τ is fixed to 1.

E. Localization Method

The attention mechanism generates the scores by computing the compatibility of the context vectors which contain features extracted by CNN from raw inputs and the hidden states of current step, indicates the scores should be high if and only if the corresponding parts contain the dominant data category. Taking advantage of this point, one can determine the locations of the target activity in a long sequence of the sensor data.

However, the scores generated by the deterministic attention are difficult to be applied in determining locations because the peak of the scores is unstable as discussed in our previous work [17]. Thus, a localization method is introduced to ameliorate it. As indicated above, we have a set of weighted score $A_t = \{\alpha_{t1}, \alpha_{t2}, \dots, \alpha_{tL}\}$, where α_{ti} is the specific weighted score of the i -th temporal location of the step t . A varied width slide window is used to sum up the score within a partial

segment

$$s_{ti} = \begin{cases} \sum_{j=1}^{i+\frac{w}{2}} \alpha_{tj} & \text{for } i < \frac{w}{2} \\ \sum_{j=i-\frac{w}{2}}^{i+\frac{w}{2}} \alpha_{tj} & \text{for } \frac{w}{2} \leq i \leq n - \frac{w}{2} \\ \sum_{j=i-\frac{w}{2}}^n \alpha_{tj} & \text{for } i > n - \frac{w}{2} \end{cases} \quad (9)$$

where the localization score s_{ti} is corresponding to the summation of the weighted score around the temporal location i . The range of this calculation is equal to the slide window width, which is varied as the temporal location i changes. The maximum of window size w is defined as

$$w = \frac{f}{p^s} * d \quad (10)$$

where the f is the sampling rate and the p is pooling size. The s is the number of pooling layers and the d is the average duration of the activities. In our implementation of the OPPORTUNITY dataset, the pooling size is 2, and the number of pooling layers is 3. Besides, according to ground truth label, the average duration of the activities is about 3 seconds. Consequently, the maximum of window size w is set to 12. The total temporal location n is equal to the length of the set of weights score A_t . Then we normalize the s_{ti} into $[0,1]$

$$\bar{s}_{ti} = \frac{s_{ti} - \min_i s_{ti}}{\max_i s_{ti} - \min_i s_{ti}}. \quad (11)$$

We denote the \bar{s}_{ti} as normalized localization score which presents the importance of the location i . Finally, the locations with scores larger than threshold are labeled as potential activity of interest. The hyper parameter (i.e., threshold value) is tuned in an empirical way. In our case, the threshold value is set to 0.7, which can better demonstrate the visualizing location results.

V. EXPERIMENTS

The effectiveness of the proposed model is examined on the three public HAR datasets: the UniMiB-SHAR dataset, our collected SWLM dataset and the OPPORTUNITY dataset. The former is to validate whether our model has the capacity to implement traditional HAR tasks. The last two are to explore the performance of the proposed model on sequential weakly supervised HAR task.

The experiments are performed on a workstation with CPU Intel i7 6850 k, 64 GB memory, and a NVIDIA GPU 1080ti with 11 GB memory. All algorithm is implemented in Python by using the deep learning framework TensorFlow. In the experiments, the number of epoch is set to 100 and Adam optimization method is used to train our model. The learning rate is set to 0.00 025. Despite different sampling rates, for simplicity and without loss of generality, the constant batch size of 50 is used across all the datasets. During the training, four-fifths of the training set is used to train model, and one-fifth is used as validation set to tune the hyperparameters and save model with the best performance. The test set is used to evaluate the trained model.

A. Experiments on UniMiB-SHAR Dataset

We compare the experimental results of our model and the baseline CNN model on the UniMiB-SHAR dataset in the metric

TABLE I
EXPERIMENT ON UNIMIB-SHAR DATASET

Model	Accuracy
CNN	72.5%
RAN	72.8%

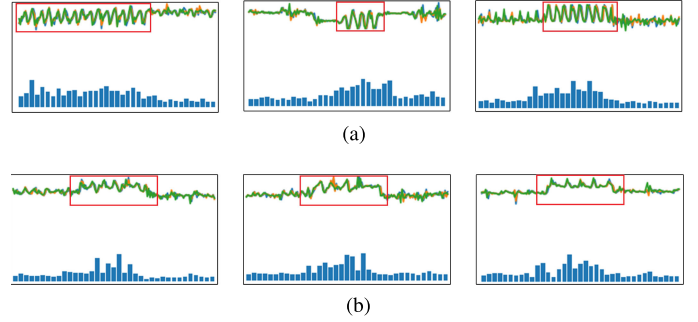


Fig. 5. Some examples of experiments on UniMiB-SHAR dataset. (a) "Running." (b) "Going down."

of classification accuracy. In this experiment, the baseline CNN consists of three convolutional layers with 32, 64, 128 kernels and two max pooling layers between these three convolutional layers. One fully connected layers with 100 units is used, and then a softmax layer is used to output the classification results. The baseline CNN without the fully connected layer is corresponding to the features extractor of our model.

The results are shown in Table I. Our model and the baseline CNN perform comparably well with regard to traditional task that belongs to supervised learning. Our RAN model replaces the final fully connected layer of the fundamental CNN architecture with the attention module. As shown in Fig. 5(a), the attention module can identify the salient activity data areas and enhance their influence, meanwhile suppressing the irrelevant and potentially confusing information in other activity data areas. In each subfigure of Fig. 5, the samples of sensors data and attention scores are drawn by colored lines and bars respectively. For each sample, the attention mechanism tends to highlight the target activity. The higher bars indicate higher attention scores, which correspond to the regions with red rectangles where the target activity occurs. However, compared with the attention mechanism proposed in [17], [26], the soft attention only adds the weights to the features extracted at the end of the CNN pipeline, which potentially weakens the specific features selection capability of attention mechanism. Thus, for this dataset, our model is not superior to the traditional methods, but still performs satisfactorily.

B. Experiments on SWLM Dataset

In this experiment, we use a CNN introduced in Section IV as feature extractor. A fully connected layer and a softmax layer are added to build the baseline CNN. The five different experiments are designed according to the types of activities which appears in training set and test set, as shown in Table II. Throughout above experiments, we can validate the effectiveness of the RAN model

TABLE II
EXPERIMENTS CASES AND RESULTS

Case	Distribution		Accuracy	
	Train	Test	CNN	RAN
1	A, B, C	A, B, C	100%	100%
2	A-B, B-A, C-A,	A-B, B-A, C-A,	*99.2%	99.0%
	A-C, B-C, C-B	A-C, B-C, C-B		
3	A, B, C,	A, B, C,	*98.9%	98.5%
	A-B, B-A, C-A,	A-B, B-A, C-A,		
	A-C, B-C, C-B	A-C, B-C, C-B		
4	A, B, C	B-A, C-A, C-B	-	*85.6%
	A-B, A-C, B-C			
5	A, B, C, A-B	B-C, C-B	-	-
	B-A, A-C, C-A			

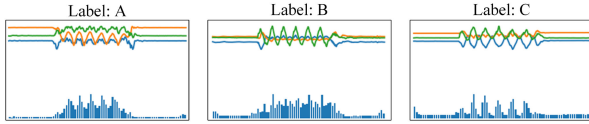


Fig. 6. Examples of experiment case 1. The attention mechanism of our model weights the features of different locations.

in HAR tasks, and explore its potential applications for activity localization tasks.

Case 1: In this case, the training set and test set only contain the sensor data samples consisting of one weakly labeled activity, which can be seen as a traditional recognition task. Due to the simple constitution of the activities, it is easy for our RAN model and the baseline CNN to extract distinct features from the dataset. So both methods can achieve an almost 100% accuracy on this traditional task. Moreover, the attention mechanism of our model exerted on the dataset is shown in Fig. 6. Note that the purpose of our work is not to pursue higher performance in recognizing activity, but to develop a sequential activity recognition and localization method which can detect and locate accurately each activity in one sample, as mentioned in the following experiments.

Case 2: This case is to test whether our model can detect activities in situation that the samples of training set contain multiple types of labels. The baseline CNN is used to perform the sequentially weakly labeled multiactivity recognition task. However, as we have known, the baseline CNN only can be trained by the samples with one label. Actually, the baseline CNN is unable to respectively extract features of multiple activities with different labels simultaneously. In order to make comparison, we compel the baseline CNN to implement the recognition task by annotating the multiactivity samples as a new label (e.g., marking “A-B” as “D,” “B-A” as “E,” etc.). As a result, it degenerates into a traditional supervised learning task, which aims to classify six types of activities. Using this idea, the baseline CNN can obtain a 99.2% classification accuracy, but the annotation processes become much harder. Thus, we mark the result with asterisk “*”. Actually, the baseline CNN

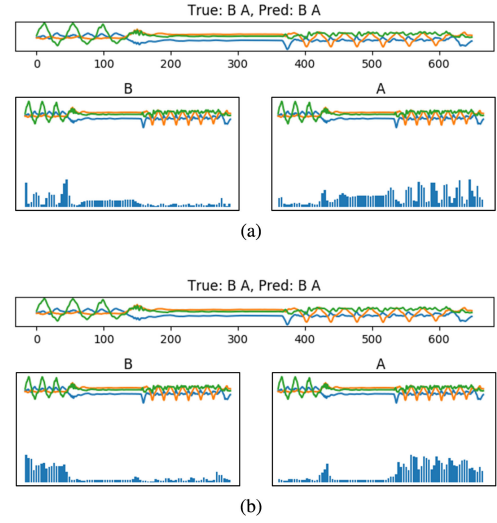


Fig. 7. (a) Examples of experiment case 2. At each step, the model produces an attention map revealing where the current step is focusing on and an classification result conditioned on the focused part. (b) Examples of experiment case 3. The attention maps become more distinct.

does not recognize every activity contained in one segment, but classifies simply this sample as a whole. On the contrary, the RAN model at one time only use the features of an activity in the sequence to perform recognition. Therefore, the RAN model sometimes shows a worse accuracy compared to the baseline CNN. However, the advantage of our model lies in that it can locate every labeled activity in one sample, and in the meantime get a satisfactory classification result.

Fig. 7(a) shows the recognition examples of our RAN model. At each step, the model produces an attention map revealing where the current attention is focusing on and a classification result corresponding to the focused activity simultaneously. In spite of almost the same classification accuracy, our model is very different from the baseline CNN, which treats these multi-activity samples as one whole. Our RAN model can recognize every labeled activity in one sample, and in the meantime get a satisfactory classification result. We stress that our RAN model can be used to handle the weakly labeled sequential activity dataset, which differs completely from the traditional supervised learning techniques requiring the accurate bounding boxes for annotating the training dataset. Thus, the complex and laborious process of manual annotation work can be greatly alleviated by our model, which can automatically crop the desired activity by utilizing the recurrent attention mechanism to impose the proper weights on the sequential activity data. That is to say, we can determine the accurate location of every labeled activity by visualizing the attention maps. However, as shown in Fig. 7(a), the attention maps can not track the location of the desired activity very well, which will be solved in the next case.

Case 3: For the case 2, the attention maps can not match accurately the location of every activity in one sample. As the samples in the dataset only contain the multiple labeled activities, the attention module can not learn the features of single activity. In this case, we combined the data samples of single activity (i.e., “A,” “B,” and “C”) and multilabeled activity (i.e., “A-B,” “B-A,”

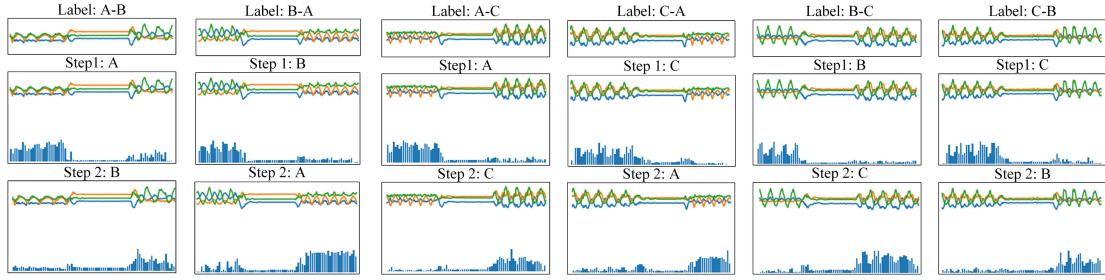


Fig. 8. Some additional examples of experiment case 3. The attention module has better weighting capability and the generated attention maps become more clear, due to the addition of single activity.

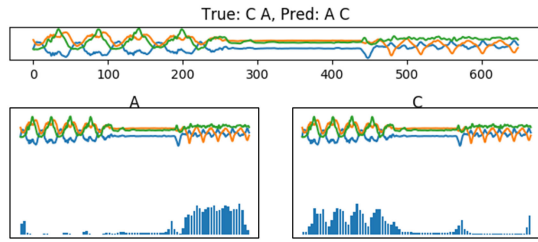


Fig. 9. Example of experiment case 4: The model can recognize the activities with correct classification results but reverse order.

“A-C,” “C-A,” “B-C,” and “C-B”) into the dataset, in order to further explore the effectiveness of our model. Table II indicates that the baseline CNN and our RAN model still can obtain 98.9% and 98.5% classification accuracy respectively, which are almost the same. The baseline CNN shares the same strategy as case 2 to perform the multiactivity recognition task, thus we mark the result with “*”. Fig. 7(b) shows that compared to the case 2, the attention module has better weighting capability and the generated attention maps become more clear, due to the addition of single activity. As shown in Fig. 8, more clear maps facilitate the determination of activity location, which makes it possible to automatically crop the regions of interest for acquiring the labeled HAR dataset by roughly marked the sequential activity data. The location part will be discussed in Section V-D.

Case 4: We further try to test whether our model can recognize the sequential activity which has never occurred in the training set. In this case, a relatively simple recognition task is proposed. We perform the recognition for sequential activity of reverse order, saying that the training set contains “A,” “B,” “C,” “A-B,” “A-C,” and “B-C” while the test set contains “B-A,” “C-A,” and “C-B.” Fig. 9 shows that the attention can focus on the right location of the current activity at each step, but the LSTM submodule, in charge of generating captions for sequential activity, outputs sequential classification results with reverse order. After labeling reversely the sequential activities of classification results, we still can obtain a classification accuracy of 85.6%, which is marked with “*”. The results indicate that our model can recognize the sequential activity with reverse order, which never exists in the training set.

Case 5: In this case, the dataset is reorganized as follows: the training set contains “A,” “B,” “C,” “A-B,” “B-A,” “A-C,” and “C-A,” and the test set contains “B-C” and “C-B.” We continue to

TABLE III
EXPERIMENT ON OPPORTUNITY DATASET

Model	CNN	[14]	[17]	RAN
-Strictly Labeled-				
Accuracy	77.8%	82.6%	78.8%	78.6%
-Sequential Weakly Labeled-				
Accuracy	-	-	-	77.5%

explore whether our model can recognize the sequential activity which never appears in the training set. The case 4 can be seen as a special case, where the reverse order condition holds. The result indicates that the data samples of “B-C” and “C-B” can not be recognized and our RAN model fails to generate new captions for sequential classification results. Actually, the LSTM submodule can not remember the sequential activity which never appears at the training stage, and the attention module does not learn how to impose the weights on these samples. That is to say, to realize accurate annotation via the RAN model, the desired sequential activities have to be roughly segmented and trained in advance.

On the whole, the above cases indicate that, unlike the baseline CNN which can only handle traditional supervised learning task, our RAN model is able to recognize and locate every activity contained in a long sequence and achieves an almost the same classification accuracy with the CNN. In other words, attention can tell where to focus along a long sequence of sensor data. Our RAN model can aid the annotator to perform annotation via skimming through raw sensor data and effectively label all activity instances. The only possible obstacle to our model is to recognize the sequential weakly labeled data samples which never appears at the training stage. This is one common problem for supervised learning techniques, which can be easily solved by roughly segmenting and training the desired sequential activity in advance for the RAN model. In addition, our model can recognize the data samples with reverse order label, which indicates the limitation of LSTM does not impede the implementation of the attention.

C. Experiments on OPPORTUNITY Dataset

We compare our model with general baseline CNN and DeepConvLSTM [14], as well as attention-based CNN [17] on

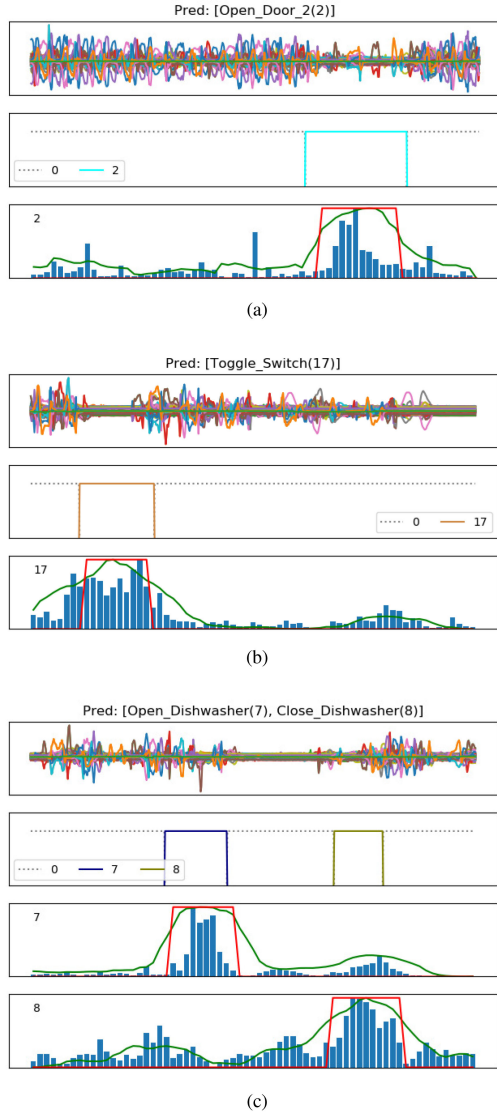


Fig. 10. Visualizing some samples of localization experiments on the OPPORTUNITY dataset. Each subfigure contains the sensor data diagram, the ground truth and the localization results from top to bottom. The tag “0” is referred to background activity.

the public benchmark OPPORTUNITY dataset, which demonstrates our model’s general ability. The design of CNN is the same to the above SWLM experiment in which the structure of CNN is equal to the feature extractor of our RAN model. Furthermore, the CNN is modified by attention mechanism mentioned in [17] at final convolutional layer to build the attention-based CNN. The DeepConvLSTM model is rebuilt according to the original paper [14]. Table III shows that, in strictly labeled case, our RAN model cannot outperform the start-of-art DeepConvLSTM model and the attention-based CNN, due to its simplicity of feature extractor CNN. However, our RAN model is superior to the baseline CNN. In addition, this method is very applicable for recognizing and locating sequential weakly labeled data. Although we may compel the traditional methods such as CNN and DeepConvLSTM to implement recognition task by annotating the multiactivity data samples as a new label

like the last experiment did, it will greatly increase the burden of sensor data annotation, not to mention the numerous combinations of activities in the OPPORTUNITY dataset. Besides, the experiment result of sequential weakly labeled data recognition on SWLM dataset is superior to that of OPPORTUNITY dataset, because the background activity of the latter is more confused and the adjacent activities contained in some samples are too close. Nevertheless, the experiment result demonstrates that our RAN can obtain 77.5% classification accuracy on sequential weakly labeled data, which reaches the level that the CNN performing traditional recognition task.

D. Localization Experiments

By utilizing the weighted scores produced by the RAN model, we are able to locate the sequential weakly labeled activity, as shown in the above experimental figures. Moreover, we convert the weighted score to the normalized localization score on sequential weakly labeled sample of OPPORTUNITY dataset. As shown in Fig. 10, each subfigure contains the sensor data diagram, the ground truth, and the localization results from top to bottom. The figures show that compared with the weighted score plotted in blue, the green curve of the normalized localization score is more beneficial to locate the labeled activities, because the green curve concentrates on the peak point where the target activity happens more intensively. The red curve is used to mark the partial segment where the normalized localization scores are above the threshold value. The red crops can be regarded as the regions of interest, which are close to the scale of ground truth. In addition, it can be seen that the RAN model can not only locate the one activity of interest in weakly labeled samples as shown in Fig. 10(a) and (b), but also deal with the sequential weakly labeled sample whose segment contains multiple activities as shown in Fig. 10(c).

VI. CONCLUSION

One challenge for deep HAR recognition is the collection of annotated or “ground truth labeled” training data. Ground truth annotation is an expensive and tedious task, in which the annotator has to perform annotation via skimming through raw sensor data and manually label all activity instances. However, the time series data recorded from multimodal embedded sensors such as accelerometer or gyroscope are far more difficult to interpret than data from other sensors modalities, such as cameras. It requires laborious human efforts to accurately segment and label a target activity from a long sequence of time series sensor data. Therefore, it deserves further research whether we can directly recognize and locate one or multiple target activity from coarsely labeled sensor data. That is to say, our main research motivation is to simultaneously infer multiactivity types from the coarse-grained sequential weak labels and determine specific locations of every target activity with only knowledge of which types of activities contained in the long sequence. It will greatly reduce the burden of manual labeling.

Hence, we develop a RAN model for multiactivity recognition that can repeatedly pay attention to the activity of interest at each step. The experiments show that our RAN model can perform

HAR tasks on sequential weakly labeled data that the traditional deep learning methods can hardly handle. Besides, the RAN model provides a way to locate the activities of interest on the sensors data. Our research also provide a feasible way to apply the attention mechanism to aid to the location and annotation of motion sensor data. Actually, the location of motion sensor data has seldom been researched. Attention mechanism demonstrates its superiority in recognition and location task for motion sensor data, which deserves deeper studies. For example, building a more efficient feature extractor subnetworks for the RAN model and attention mechanism. We put these as our future works.

REFERENCES

- [1] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, 2014, Art. no. 33.
- [2] T. Magherini, A. Fantechi, C. D. Nugent, and E. Vicario, "Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 6, pp. 509–521, Nov. 2013.
- [3] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Trans. Syst., Man, Cybern., Part A*, vol. 39, no. 5, pp. 949–959, Sep. 2009.
- [4] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.
- [5] L. Pei *et al.*, "Human behavior cognition using smartphone sensors," *Sensors*, vol. 13, no. 2, pp. 1402–1424, 2013.
- [6] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. Int. Conf. Pervasive Comput.*, 2004, pp. 1–17.
- [7] T. Huynh, U. Blanke, and B. Schiele, "Scalable recognition of daily activities with wearable sensors," in *Proc. Int. Symp. Location-Context-Awareness*, 2007, pp. 50–67.
- [8] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newslett.*, vol. 12, no. 2, pp. 74–82, 2011.
- [9] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Int. Workshop Ambient Assist. Living*, 2012, pp. 216–223.
- [10] M. Zeng *et al.*, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. IEEE 6th Int. Conf. Mobile Comput., Appl. Serv.*, 2014, pp. 197–205.
- [11] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *Proc. IEEE Int. Conf. Systems, Man, Cybern.*, 2015, pp. 1488–1492.
- [12] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1307–1310.
- [13] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Conf. Artif. Intell.*, 2015, pp. 3995–4001.
- [14] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016, Art. no. 115.
- [15] P. Kumar, S. Mukherjee, R. Saini, P. Kaushik, P. P. Roy, and D. P. Dogra, "Multimodal gait recognition with inertial sensor data and video using evolutionary algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 5, pp. 956–965, May 2019.
- [16] J.-H. Kim, G.-S. Hong, B.-G. Kim, and D. P. Dogra, "Deepgesture: Deep learning-based gesture recognition scheme using motion sensors," *Displays*, vol. 55, pp. 38–45, 2018.
- [17] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities recognition with wearable sensors," *IEEE Sensors J.*, vol. 19, no. 17, pp. 7598–7604, Sep. 2019.
- [18] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.
- [19] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.
- [20] C. A. Ronao and S.-B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *Proc. Int. Conf. Neural Inf. Process.*, 2015, pp. 46–53.
- [21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.
- [22] Q. Teng, K. Wang, L. Zhang, and J. He, "The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7265–7274, Jul. 2020.
- [23] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, 2019.
- [24] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.
- [25] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, and I. Lane, "Semi-supervised convolutional neural networks for human activity recognition," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2017, pp. 522–529.
- [26] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention," *arXiv:1804.02391*, 2018. [Online]. Available: [online] Available: <http://arxiv.org/abs/1804.02391>
- [27] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2127–2136.
- [28] J. He, Q. Zhang, L. Wang, and L. Pei, "Weakly supervised human activity recognition from wearable sensors by recurrent attention learning," *IEEE Sensors J.*, vol. 19, no. 6, pp. 2287–2297, Mar. 2019.
- [29] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzec, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 2, 2018, Art. no. 679.
- [30] N. Kawaguchi *et al.*, "HASC challenge: Gathering large scale human activity corpus for the real-world activity understandings," in *Proc. 2nd Augmented Hum. Int. Conf.*, 2011, pp. 1–5.
- [31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [32] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014, pp. 1–13.
- [33] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: <http://arXiv:1409.0473>