DeepSeg: Deep Learning-based Activity Segmentation Framework for Activity Recognition using WiFi

Chunjing Xiao, Yue Lei, Yongsen Ma, Fan Zhou, and Zhiguang Qin

***Abstract*—Due to its non-intrusive character, WiFi Channel State Information (CSI)-based activity recognition has attracted tremendous attention in recent years. Since activity recognition performance heavily relies on activity segmentation results, a number of activity segmentation methods have been designed, and most of them focus on seeking optimal thresholds to segment activities. However, these threshold-based methods are strongly dependent on designers' experience and might suffer from performance decline when applying to the scenario including both fine-grained and coarse-grained activities. To address these challenges, we present DeepSeg, a deep learning-based activity segmentation framework for activity recognition using WiFi signals. In this framework, we transform segmentation tasks into classification problems and propose a CNN-based activity segmentation algorithm, which can reduce the dependence on experience and address the performance degradation problem. To further enhance overall performance, we design a feedback mechanism, where the segmentation algorithm is refined based on the feedback computed using activity recognition results. The experiments demonstrate that DeepSeg acquires remarkable gains compared with state-of-the-art approaches.**

***Index Terms*—Channel state information, activity recognition, segmentation, time-series, change point detection.**

I. Introduction

Human activity recognition is considered a key aspect for many emerging Internet of Things applications, such as smart homes and health care. For these applications, various kinds of activities need to be accurately recognized to contribute to the quality of life [1]. And a number of human activity recognition methods have been designed with various techniques, such as wearable sensors [2], [3], smart phones [4], [5], and cameras [6], [7]. Recently, activity recognition based on WiFi Channel State Information (CSI) has been attracting tremendous attention due to its ubiquitous availability and non-intrusive character. By collecting CSI, many works have explored recognition of different kinds of behaviors, such as fine-grained activities (e.g. gestures [8] and sign language [9]) and coarse-grained activities (e.g. walking [10] and falling [11]).

For CSI-based activity recognition, after data collection and pre-processing, the remaining procedure can mainly be divided into two stages: activity segmentation and activity classification [12], [13]. The former aims at detecting the start and end points of activities on continuously received CSI streams. Activity classification concentrates on classifying activity segmentation results into corresponding categories. As inputs of activity classification, the quality of activity segmentation has a significant impact on the performance of activity classification. Therefore, a large number of studies have been initiated on activity segmentation. For CSI data, because fluctuation ranges of CSI amplitudes when activities occur are much larger than that when no activity presents, most existing works focus on designing threshold-based segmentation methods, which attempt to seek an optimal threshold to detect the start and end of an activity, such as Tw-See [14] and Wi-Multi [15]. If the fluctuation degree of CSI waveforms exceeds this threshold, an activity is considered to happen.

In spite of the success of these threshold-based segmentation methods, there still exist some weaknesses. First, policies of noise removal and threshold calculation are usually determined based on subjective observations and experience, and some recommended policies might even be conflicted. For example, both CARM [16] and WiAG [17] suggest Principal Component Analysis (PCA) can be used to effectively remove the noise of CSI streams, but different principal components are s-elected for further processing. CARM [16] recommends to use the second and third principal components and discard the first one. However, WiAG [17] only adopts the third component because the authors observe that the second component also captures some noise. Instead, Tw-See [14] shows that PCA is not appropriate for the scenario of WiFi signals through walls and the opposite robust PCA can be used to remove noise. Meanwhile, the first principal component is recommended for activity segmentation.

Second, threshold-based segmentation methods may suffer from significant performance degradation when applying to practical scenarios. Existing works mainly focus on either fine-grained or coarse-grained activities and achieve expected performance. However, under practical scenarios, fine-grained and coarse-grained activities can alternately and randomly

Manuscript received August 14, 2020; revised September 16, 2020; accepted October 08, 2020. This work is supported by the National Natural Science Foundation of China (No.61402151 and 61806074), Science and Technology Foundation of Henan Province of China (No.162102410010). *(Corresponding author: Fan Zhou.)*

Chunjing Xiao is with the Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China, and also with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: chunjingxiao@gmail.com).

Yue Lei is with the School of Computer and Information Engineering, Henan University, Kaifeng 475004, China.

Yongsen Ma is with the Computer Science Department, College of William & Mary, Williamsburg, VA 23187, USA.

Fan Zhou, and Zhiguang Qin are with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: fan.zhou@uestc.edu.cn)

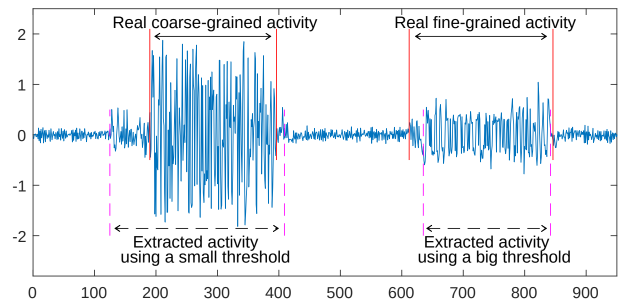


Fig. 1: Performance of threshold-based activity segmentation methods for mixed activities. A small threshold is appropriate for the fine-grained activity but not for the coarse-grained one. And the reverse is true for a big threshold.

occur. We call the mixture of both kinds of activities as mixed activities. And, the optimal threshold for fine-grained (coarse-grained) activities might not be proper for coarse-grained (fine-grained) ones because of the enormous difference between their CSI amplitudes. An example from our collected data is shown in Figure 1. This figure presents the first-order differences of CSI amplitudes of one subcarrier for one coarse-grained activity (left) and one fine-grained activity (right). Here the solid red lines are the real start and end points of activities, and dotted pink lines are the ones detected by a given threshold. Under this case, if a small threshold is selected to identify the start and end of activities, the fine-grained activity can be accurately extracted, but it is not appropriate for the coarse-grained activity because a little noise data near this activity will mistakenly be regarded as activity data. On the contrary, a big threshold is appropriate for the coarse-grained activity but not for the fine-grained one. Therefore, it is difficult to set a threshold to accurately segment all kinds of activities. Also, our experiments indicate that threshold-based methods always obtain lower segmentation accuracy on mixed activities than either of them. Hence, the performance of the threshold-based methods may severely decline when applying to mixed activities.

Third, activity segmentation and activity classification, which are closely interrelated, are usually treated as two separate states. Existing studies generally first conduct activity segmentation and then classify segmented activity data into corresponding categories, and fail to consider their connection. While, to a large extent, classification accuracy can reflect the quality of segmentation results. Hence, the results of activity classification can be used as the feedback to refine activity segmentation algorithms for enhancing the overall performance.

Apart from threshold-based segmentation methods, change point detection techniques [18] can be used to conduct activity segmentation. However, these techniques generally adopt a threshold value to determine activity boundaries and selecting the optimal threshold value is very difficult [19]. They are still confronted with these problems, such as suffering from performance decline for mixed activities and treating activity segmentation and activity classification as two separate states.

In this paper, we present a deep learning-based activity segmentation framework, DeepSeg, to address these problems. In this framework, to avoid experience-dependent noise removal and threshold calculation and address performance decline for mixed activities, we transform activity segmentation tasks into classification problems and propose a CNN-based activity segmentation algorithm to identify the start and end points of activities. This algorithm first discretizes continuous CSI streams into equally sized bins, and then our designed state inference model (a CNN architecture) is adopted to classify these bins into four states: *static-state*, *start-state*, *motion-state* and *end-state*. Third, the start and end points of activities are identified based on these state labels and the activity data is extracted. Last, the segmented activity data is fed into the activity classification model for activity recognition.

To enhance overall performance, we propose a feedback mechanism between the activity segmentation algorithm and the activity classification model. Inspired by the idea of reinforcement learning [20], we use the probability distribution generated by the activity classification model to compute the concentration degree as the feedback for refining the activity segmentation algorithm. For a given sample, its concentration degree, the combination of its probability entropy and maximum probability, can reflect the confidence of it belonging to the corresponding category. Hence, high confidence indicates that it is an expected sample for activity recognition. Because high-quality samples will be helpful to enhance the classification performance [21], we tune the state inference model by increasing the weights of samples with higher confidence. As a result, the adjusted segmentation algorithm can output more expected samples for activity classification, and the overall performance can be enhanced.

We summarize the main contributions of this paper as follows:

- We propose a deep learning-based activity segmentation framework, DeeSeg, which can reduce the dependence on experience and address performance degradation for mixed activities.
- We design a feedback mechanism between the segmentation algorithm and the classification model, which can enhance overall performance by jointly training them.
- Experiment results illustrate that our framework outperforms the state-of-the-art approaches in scenarios with a single kind of activities or mixed activities. The data and code are available online[1].

II. Related Work

For activity recognition, some researches propose activity segmentation algorithms. And others assume the activities have been segmented and only focus on activity recognition. Here, we illustrate an overview about the most closely related researches in these two aspects and highlight the major differences between our work and these studies.

A. Activity Segmentation

For WiFi CSI-based activity recognition we focused on, to efficiently and accurately classify activities, existing works

[1]https://github.com/ChunjingXiao/DeepSeg

usually first identify activity boundaries and extract activity data, and then conduct activity classification [12], [22]. There are mainly two ways for implementing this kind of segmentation: threshold-based methods and Change Point Detection (CPD)-based methods. Here, we present a brief overview about these two ways.

Threshold-based activity segmentation. For CSI data, there is a distinct character that the variance of CSI amplitudes in the presence of activities is much bigger than the one in the absence of activities. Hence, studies about CSI-based activity recognition generally adopt thresholds to detect the start and end points of activities and extract activity data [13], [12].

Some researchers utilize a *fixed threshold* to segment activities. They attempt to seek optimal policies of noise removal and threshold selection based on their observations for activity segmentation. For example, Virmani and Shahzad [17] found that PCA can be adopted to effectively remove the noise of CSI streams, and a fixed threshold can be used to efficiently distinguish gesture data from CSI streams. And they adopted the second principal component for gesture segmentation. Wu *et al.* [14] designed a normalized variance sliding window algorithm for activity segmentation to meet scenarios of the WiFi signals through a wall. They demonstrated that PCA is not appropriate for their experimental data. Therefore, they introduced the opposite robust PCA to reduce the interference and noise of CSI streams and adopted the first principal component for activity segmentation. Sheng *et al.* [23] presented a segmentation algorithm based on the difference of time series. This algorithm adopts their designed mean absolute differences and the pre-set start threshold and end threshold to detect the activity part. Bu *et al.* [24] adopted an optimal threshold computed based on the training set to identify the start and end of activities, and proposed a motion pause buffer algorithm to address incorrect gesture segmentation caused by some short pauses during a gesture. Wang *et al.* [25] proposed a two-phase segmentation algorithm to distinguish the fall activity. In this algorithm, a threshold-based sliding window approach is adopted to identify whether CSI waveforms are in the fluctuation state or stable state. And then an appropriate trace back window size from the end point is utilized to detect the starting point of the fall or fall-like activity.

Other works adopt a *dynamic threshold* to segment activities, which means that the threshold can be adjusted according to the noise level and experimental environment. For instance, Feng *et al.* [15] designed a two-step algorithm to detect the start and end points of activities for the scenario with multiple subjects. In the first step, a given CSI stream is split into even slots where the slot length should be changed according to specific scenarios. And the variances of CSI amplitudes in each slot are stacked as an array in ascending order. The average difference between the sum of the largest half values and the sum of the smallest half values in this array are regarded as the threshold. In the second step, the largest eigenvalues from both amplitude and calibrated phase correlation matrices are calculated to eliminate potential false detection. Wang *et al.* [26] adopted two thresholds to detect the start and end of an activity. They utilized an activity amplitude threshold to find potential activities from the denoised CSI streams, and an activity duration threshold to eliminate a short wave burst in CSI streams which can be viewed as noise. The activity amplitude threshold is computed based on CSI streams when persons performing activities and ones when persons keeping stationary. And this threshold can be iteratively updated using the new collected CSI data to better fit the environment change. Yan *et al.* [27] proposed an adaptive activity cutting algorithm based on the difference in signal variance between the action and non-action parts. This algorithm updates the threshold based on the changes of CSI waveforms in the current sliding window to achieve the trade-off between performance and robustness. Wang *et al.* [28] presented a dynamic thresholding algorithm for activity segmentation. They set the detection threshold as three times the noise level and updated the noise level estimation using an Exponential Moving Average algorithm during the silent period.

The above researches focus on either fine-grained activities or coarse-grained activities and achieve expected performance. However it is difficult to seek optimal thresholds for mixed activities because of the great difference between CSI amplitudes. And it is laborious and subjective to manually seek appropriate policies for noise removal and threshold calculation. Note that the dynamic threshold in these studies cannot address the performance degradation problem of mixed activities because it means that the threshold can change according to the experimental environment and noise level. And the threshold will be a relatively fixed value under the same environment. For example, the dynamic threshold in [15] is completely dependent on the sampling rate (which is the ratio of the CSI series length to the slot length). For a given CSI series consisting of mixed activities, the same threshold is used to segment fine-grained activities and coarse-grained activities. Here, we adopt a deep learning model to segment activities, which does not require noise removal and threshold calculation, and can address the performance decline problem for mixed activities.

CPD-based activity segmentation. Change point detection (CPD) is the problem of finding abrupt changes in data when a property of the time series changes, and widely used for a broad range of real-world problems such as medical condition monitoring, climate change detection, speech recognition and image analysis [18], [29]. In terms of CPD-based activity segmentation, activity breakpoints or transitions in observed sensor data can be formulated as change points, and activities can be segmented according to these change points [19].

While CPD is a thoroughly-investigated topic, only a few unsupervised methods are suitable for activity segmentation to meet the real-time requirement [30]. Specifically, Ni *et al.* [31] introduced the multivariate online change detection algorithm to detect transitions by comparing statistical values such as mean and variance-covariance matrices for two consecutive sensor data windows. Alam *et al.* [32] applied the Relative unconstrained Least-Squares Importance Fitting (RuLSIF) algorithm to accelerometer data collected from a wearable smart earring to detect gesture changes. Chakar *et al.* [33] proposed a AR1seg approach for estimating change-points in the AR(1) model based on the assumption of a common autocorrelation coefficient for all segments, and presented an exact maximum likelihood solution for parameter computation. Also maximum likelihood estimation algorithm is used to segment gestures and head-movement events [34], [35]. Recently, density ratio-based techniques are enhanced to conduct change point detection by employing new probability metrics and more sensitive change scores, and change points are determined by comparing the score with a threshold value [36]. And these techniques are further extended for activity segmentation and achieve expected segmentation performance [37].

Although CPD techniques have achieved great success in many applications, there exist limitations when applying to our scenarios. The CPD techniques generally compare change scores with a threshold value to determine whether change occurs or not, and selecting the optimal threshold value is difficult because these values may be application dependent and change over time [19]. Hence, these techniques still adopt threshold values to segment activities and suffer from the same problems to threshold-based methods, such as distinct performance decline for mixed activities and negligence of the connection between segmentation and recognition. In contrast, our proposed DeepSeg can enhance their connection and the recognition performance.

B. Activity Recognition

According to activity levels, studies on WiFi CSI-based activity recognition mainly fall into two genres: fine-grained and coarse-grained activity recognition. The former denotes behaviors performed at micro levels, such as keystrokes, sign language and gestures. For example, WiFi signals are used to identify the typed keys on a keyboard of the laptop [38] and infer sensitive keystrokes on mobile devices [39]. And SignFi [9] and WiHear [40] are designed to recognize sign language gestures and detect people speech based on the fluctuation of CSI caused by mouth movements, individually. In addition, there are many WiFi CSI-based gesture recognition systems to facilitate human-computer interaction in numerous smart home applications [8], [41], [42]

Coarse-grained activities denote behaviors performed at macro levels, such as walking, running, sitting down. For instance, a lot of researches aim to recognize daily activities to detect behavioral changes relative to health and provide an active input for the smart systems [43], [44]. The works [45], [46] try to employ CSI to recognize exercise activities to help people promote physical fitness. Further, WiFi signals are also used to monitor health related activities, such as falling, to prevent potential hazard [47], [11].

These activity recognition methods generally treat activity segmentation and activity recognition as two independent stages, i.e., they just feed segmentation results into the recognition model for activity classification. Instead, we attempt to design a feedback mechanism between them to improve overall performance. To the best of our knowledge, we are the first to transform segmentation tasks into classification problems and introduce the feedback mechanism for WiFi CSI-based activity recognition.

III. DeepSeg Framework

In this section, we present the DeepSeg framework consisting of activity segmentation and activity classification. First, we present an overview with the correlation of its components. Next, we illustrate the detailed design of the activity segmentation algorithm and the activity classification model. Finally, we provide the joint training algorithm of DeepSeg.

A. DeepSeg Overview

We present the DeepSeg framework to conduct activity recognition using WiFi CSI. In this framework, to avoid experience-dependent noise removal and threshold calculation and address the problem of performance decline for mixed activities, we propose a CNN-based activity segmentation algorithm to segment activities. To enhance overall performance, we introduce a feedback mechanism which can refine the activity segmentation algorithm based on the feedback computed using activity classification results.

Figure 2 presents an illustration of how the proposed framework works. The DeepSeg framework mainly consists of the two parts: the activity segmentation algorithm and the activity classification model. For activity segmentation, the continuously received CSI streams are first discretized into equally sized bins. And then the state inference model is used to infer states of these bins. Here we define four states: static-state, start-state, motion-state and end-state. Finally, the start and end points of activities are detected based on the inferred state labels, and activity data is extracted. For activity classification, the classification model takes segmented activity data as input and outputs the probabilities of it belonging to activity classes. And these probabilities are further used to compute the concentration degree as the feedback for the state inference model. Meanwhile, based on the feedback, the state inference model is tuned by increasing the weights of samples with higher confidence.

B. Activity Segmentation

Since fluctuation ranges of different kinds of activities are quite different, it is difficult to obtain an optimal threshold for segmenting mixed activities. Therefore, threshold-based segmentation methods may suffer from distinct performance decline for mixed activities. At the same time, the results of threshold-based methods might be subjective and environment dependant by manually seeking policies of noise removal and threshold calculation. To address these challenges, instead of using thresholds to segment activities, we transform segmentation tasks into classification problems. Since deep learning algorithms are very powerful for data classification, such as text classification [48], [21] and activity recognition [49], [50], we introduce deep learning techniques and propose a CNN-based activity segmentation algorithm. The main idea is that the CSI streams are divided into equally sized bins, and a CNN model is used to predict their state labels which can indicate whether the bins contain start or end points of activities. Further, activity data is extracted according to the predicted

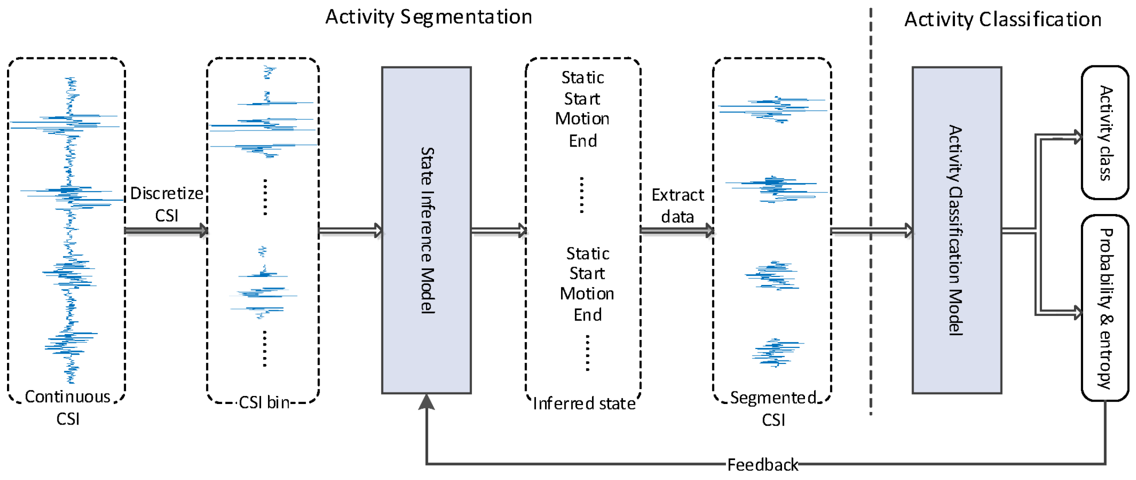


Fig. 2: DeepSeg Framework.

state labels. Our proposed segmentation algorithm needs a trained state inference model (a CNN architecture) to predict state labels of bins. Therefore, we first present how to train this model, and then illustrate the detailed algorithm.

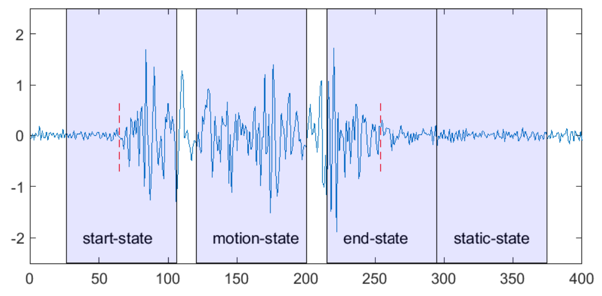


Fig. 3: Four kinds of states extracted from an activity

To train the state inference model, we extract training data from CSI streams with start and end tags of activities. Specifically, we portion continuously received CSI streams into bins with equal size, and define four state labels for these bins: *static-state*: the bin is full of CSI data in the absence of activities; *start-state*: the bin contains the start point of an activity; *motion-state*: the bin is full of CSI data in the presence of activities; *end-state*: the bin contains the end point of an activity. Figure 3 presents the examples of these four states extracted from an activity. This figure shows the first-order differences of CSI amplitudes of one subcarrier as a function of time, where the dotted red lines are the real start and end points of the activity. As shown in this figure, *start-state* contains half of the non-action part and half of the action part, and *end-state* also contains both them, but the action part is in front. Instead, *static-state* only contains the non-action part, and *motion-state* only contains the action part.

The details of extracting these bins for training the state inference model are presented in Table I. Here, t_{start} and t_{end} refer to the real start and end points of human activities respectively, and w means the window size (the length of the bin). In this table, the start point and end point for each state are shown in the third and fourth columns. For example, given an activity sample, CSI data from $t_{end} + w/2$ to $t_{end} + w/2 + w$ is labeled with *static-state*. Note that we use the first-order differences of CSI amplitudes for activity segmentation because they are more efficient than the raw amplitudes. According to this definition, each activity sample can generate four bins. And these bins are used to train the state inference model.

TABLE I: Start and end points of four states for model training

No.	Label name	Start point	End point
1	static-state	$t_{end} + w/2$	$t_{end} + w/2 + w$
2	start-state	$t_{start} - w/2$	$t_{start} + w/2$
3	motion-state	$(t_{start}+t_{end})/2-w/2$	$(t_{start}+t_{end})/2+w/2$
4	end-state	$t_{end} - w/2$	$t_{end} + w/2$

We adopt a CNN architecture as the state inference model. It can be briefly described as:

$$L = CNN(X) \tag{1}$$

where X is the input CSI data, and L is the output of the final fully connected layer. This structure is composed of convolution layers, dropout layers and max pooling layers. And the convolution parameters are W_f and b_f. The probability for state prediction $p(r|x;\Phi)$ is presented as follows:

$$p(r|x;\Phi) = softmax(W_r * L + b_r) \tag{2}$$

where W_r and b_r are parameters in the fully-connected layer and $\Phi = \{W_f, b_f, W_r, b_r\}$. Given the training set $\{X\}$, the objective function of the state inference model using cross-entropy is defined as follows:

$$\mathcal{J}(\Phi) = -\frac{1}{|X|}\sum_{i=1}^{|X|} \log p(r_i|x_i;\Phi) \tag{3}$$

The Adam algorithm [51] is adopted as the optimization method with the default hyper-parameters.

After training the state inference model, we identify the start and end points of activities according to the following three steps. (1) We divide CSI streams into bins using a w-point sliding window, where the sliding step is 1, and (2) adopt the trained state inference model to infer labels of bins. (3) Based on the labels of bins, we identify the start and end points of human activities by investigating the changes of the mode of a set of bin labels. Here the mode is the number which appears most often in the set. In other words, if the mode changes from 1 (*static-state*) to 2 (*start-state*), the corresponding bin is regarded as the start of an activity. And if it changes from 4 (*end-state*) to 1 (*static-state*), this bin is regarded as the end. Algorithm 1 presents the details about this method. The algorithm first divides input CSI streams into bins with the size w (Line 1), and adopts the trained state inference model to predict state labels of bins, which are stacked as the vector $inferred\_label$ (Line 2). After that, the algorithm tries to traverse $inferred\_label$ to detect the start and end points of human activities (Line 6). During the detection process, line 8-13 is used to determine whether the human activity starts or not. If the state mode changes from 1 (*static-state*) to 2 (*start-state*), we set $i - m/2 + 1$ as the start of an activity (Line 11), where m is the length of the list of bin labels for mode calculation. Line 13-18 is used to determine whether the human activity is over or not. If the state mode changes from 4 (*end-state*) to 1 (*static-state*), we set $i - m/2 + 1 - w$ as the end of the activity (Line 16). Here the reason of subtracting w for the end is because the current bin is full of the non-action part when the state mode becomes 1, the start point of the activities should be the beginning of this bin instead of the ending.

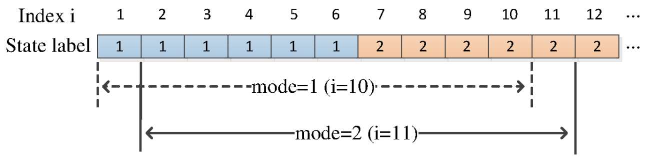


Fig. 4: An example about how to obtain the start point

For visual inspection, Figure 4 presents an example about how to obtain the start point by investigating the change of the mode, where the length of the list of bin labels for mode calculation, m, is set to 10. In this figure, when i=10, the mode of the list of state labels from index 1 to 10 is 1. When $i = 11$, the mode of the list from index 2 to 11 is 2, and the mode changes from 1 (*static-state*) to 2 (*start-state*). Note that when there are multiple values occurring equally frequently, the mode is set to the biggest of those values. Therefore, the start point, t_{start}, is set to $i-m/2+1$. When $i = 11$ and $m = 10$, t_{start} is equal to the real start point, 7, in this example. After segmenting human activities, these data will be used for activity classification.

Algorithm 1 CNN-based activity segmentation algorithm

```
Input: Trained state inference model, CSI streams, window
    size $w$, length for calculating the mode $m$
Output: Start point $t_{start}$ and end point $t_{end}$ of the activity
 1: Divide CSI streams into equal bins with the size of $w$,
    where the sliding step is set to 1
 2: Use the trained state inference model to infer labels of
    bins, and save the label sequence as $inferred\_label$
 3: $last\_mode = 0$
 4: $current\_mode = 0$
 5: $flag\_start = False$
 6: for $i = m$, ..., length($inferred\_label$) do
 7:   $current\_mode$ = the mode of the list from
      $inferred\_label_{i-m}$ to $inferred\_label_i$
 8:   if $flag\_start == False$ then
 9:     if $last\_mode == 1$ and $current\_mode == 2$ then
10:       $flag\_start = True$
11:       $t_{start} = i - m/2 + 1$
12:     end if
13:   else
14:     if $last\_mode == 4$ and $current\_mode == 1$ then
15:       $flag\_start = False$
16:       $t_{end} = i - m/2 + 1 - w$
17:     end if
18:   end if
19:   $last\_mode = current\_mode$
20: end for
```

C. Activity Classification

For CSI-based activity classification, deep learning models are generally regarded as more accurate and efficient methods because they can automatically select representative features and pre-train the models [13], [52]. Therefore, we adopt the deep learning model, CNN, to conduct activity classification, and this model takes the amplitudes of segmented CSI data as input and outputs the probability distribution over activity categories.

Apart from activity classification, the probability distribution is also used to compute the concentration degree as the feedback for the activity segmentation algorithm. For an activity sample, the concentration degree of its probability distribution can reflect the confidence of this sample belonging to the category. The high confidence suggests that it is an expected sample for activity recognition, if the predicted category is the same to the real one. Therefore, similar to works [21], [53], we adopt the confidence degree as the reward (feedback) to the activity segmentation algorithm. Based on this feedback, the activity segmentation algorithm is adjusted by forcing it to output more expected samples.

For a given sample, we use the combination of its maximum probability and probability entropy to represent the concentration degree as the feedback. Here, the maximum probability refers to the highest value among its probabilistic outputs. And the category with the maximum probability is regarded as the final classification of the input data. The probability entropy is used to reflect other probabilities besides the maximum one. Their combination can comprehensively represent the probability distribution.

Specifically, assume that $\mathbf{P} \in \mathbb{R}_{+}^{C\times 1}$ is the probabilistic output of the activity classification model for a segmented activity sample, where C is the number of activity categories. Each of its elements $p(c|x_i)$ refers to the probability that the input data x_i belongs to the category c. The maximum probability is calculated as follows:

$$M_i = \max_{c\in[1,C]} (p(c|x_i)) \tag{4}$$

For sample x_i, its probability entropy is computed as follows:

$$E_i = -\frac{1}{\ln C} * \sum_{c=1}^{C} p(c|x_i) \ln p(c|x_i) \tag{5}$$

A smaller E_i means it is easy to discriminate the input sample with more confidence. For example, when only the first item in $\mathbf{P}$ is 100% and others are 0, the probability entropy will be 0. This means that this sample will be labeled as the first category with the highest confidence. The more flat the probability distribution is, the higher the probability entropy is. And correspondingly, it is more difficult to distinguish the input sample.

Further, by combining the maximum probability and probability entropy, the feedback of sample x_i is calculated as follows:

$$D_i = \begin{cases} \alpha * M_i + (1-\alpha) * (1 - E_i), & y = y^* \\ \alpha * (1 - M_i) + (1-\alpha) * E_i, & y \neq y^* \end{cases} \tag{6}$$

where α is the parameter for adjusting the weights of the two parts, and y and y^* are the predicted and real labels respectively. If the predicted categories are equal to the real ones, higher confidence will generate bigger D_i. These values will be used as the feedback to adjust the activity segmentation algorithm for enhancing overall performance.

D. Joint Training

To improve the overall performance, we incorporate the feedback from the activity classification model into the activity segmentation algorithm, and train them jointly. We mainly apply the feedback to the state inference model. The main idea is that training samples with high confidence will play an important role in training the state inference model. The reason behind this is because increasing the weights of expected samples for model training can effectively improve classification performance [54], [21].

Specifically, by taking the feedback into account, the object function of the state inference model becomes:

$$\mathcal{J}(\Phi) = -\frac{1}{|X|} \sum_{i=1}^{|X|} \log D_i * p(r_i|x_i; \Phi) \tag{7}$$

where D_i is the confidence of the i-th sample and is computed using Equation 6. Compared with the loss function in Equation 3, this loss function considers the difference about the importance of various samples.

Considering the new loss function, we jointly train the activity segmentation algorithm and activity classification model to improve overall performance. The complete joint training process is presented in Algorithm 2. The algorithm first pre-trains the state inference model L_{seg} and the activity classification model L_{act} based on labeled data (Line 1 and 2). After pre-training, the training proceeds by iteratively updating the parameters of L_{seg} and L_{act} (Line 3-8). Specifically, Algorithm 1 is used to extract activity data from continuously captured CSI steams, and save extracted samples as $Data_{act}^{auto}$ (Line 4). And, for each sample i, its feedback D_i is computed using Equation 6 (Line 5). After that, the state inference model is refined by using the new loss function in Equation 7 (Line 6), and the activity classification model is trained based on $Data_{act}^{auto}$ (Line 7). When all the training is finished, the activity classification model will be adopted to classify activities.

Algorithm 2 Overall Training Procedure

```
Input: Labeled segmentation data ($Data_{seg}^{lab}$), labeled activity
    data ($Data_{act}^{lab}$), training epochs $N_{epoch}$.
1: Pre-train the state inference model $L_{seg}$ using $Data_{seg}^{lab}$
2: Pre-train the activity recognition model $L_{act}$ using
   $Data_{act}^{lab}$
3: for $num\_epoch$ = 0, ..., $N_{epoch}$ do
4:     Use Algorithm 1 to automatically segment activities,
       and save results as $Data_{act}^{auto}$
5:     For each sample $x_i$, use Equation 6 to generate its
       feedback $D_i$
6:     Update the parameters of $L_{seg}$ using the loss function
       in Equation 7 based on $Data_{seg}^{lab}$
7:     Update the parameters of $L_{act}$ based on $Data_{act}^{auto}$
8: end for
```

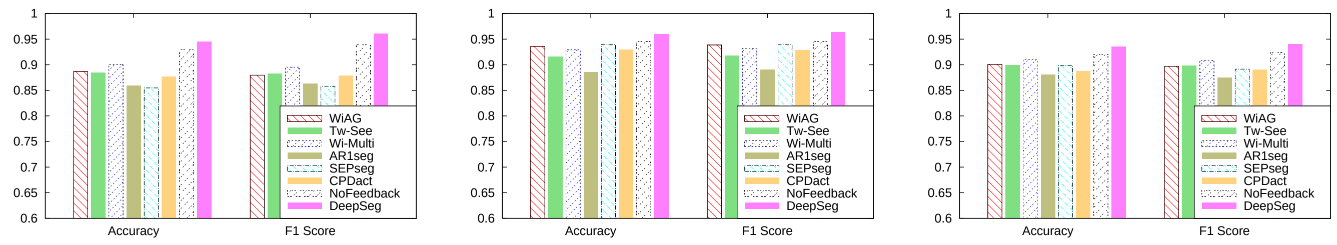
IV. Experimental Evaluation

In this section, we evaluate the effectiveness of DeepSeg on the data sets of coarse-grained, fine-grained and mixed activities. Also, the roles of classification models and training data size are investigated.

A. Experiment Setup

To collect CSI data of human activities, we deploy a commercial WiFi router and a laptop with the Intel 5300 NIC in an ordinary meeting room. Figure 6 presents the room layout for data collection. We use the router with 1 antenna as the transmitter and the laptop with 3 antennas as the receiver. And the CSI sampling rate is set to 50 packets/s. The collection tool [55] we used can acquire CSI of 30 subcarriers from each pair of transmit-receive antennas. Based on these devices, we employ 5 volunteers with different body shape and age to operate 10 kinds of activities. These behaviors are composed of 5 fine-grained activities (hand swing, hand raising, pushing, drawing O and drawing X) and 5 coarse-grained activities (boxing, picking up, running, squatting and walking). For each CSI series, the volunteers are asked to operate a fine-grained activity five times and a coarse-grained activity five times. And they are requested to take brief pauses before and after each activity. We collect 30 samples for each activity per user. As a result, there are 1,500 samples for these experiments, and 80% of them are regarded as training set and the rest as test set.

We set the hyper-parameters empirically by performing a 5-fold cross-validation on the training set. And the learning rate and the mini-batch size are set to 0.006 and 16, individually. We optimize parameter α in Equation 6 by conducting a grid search on the values between 0.1 and 1. The accuracy is not sensitive to α. Thus, we set α = 0.8 because it can obtain a slightly better result. Similarly, the window size w in Table I



(a) Mixed activity (b) Coarse-grained activity (c) Fine-grained activity

Fig. 5: Performance of activity recognition

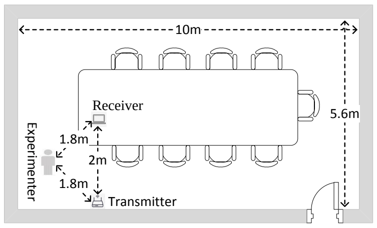


Fig. 6: The meeting room layout for data collection

is empirically set to 120. For implementation details about DeepSeg, Table II demonstrates the CNN architecture of the state inference model. In this table, Conv, T-conv, BN, WN, FM and NiN stand for Convolution, Transposed-Convolution, Batch Normalization, Weight Normalization, Feature Maps and Network in Network, respectively. The activity classification model adopts the same architecture except that the input in Line 1 is $200\times30\times3$ CSI and the stride in Line 5 and 9 is 5×2.

TABLE II: Network architecture of CNN

NO.	Operation	Configuration
1	Input	$120\times30\times3$ (CSI values)
2	Dropout	$p = 0.2$
3	Conv	Kernel=3×3 FM=96 WN lReLU
4	Conv	Kernel=3×3 FM=96 WN lReLU
5	Conv	Kernel=3×3 Stride=4×2 FM=96 WN lReLU
6	Dropout	$p = 0.5$
7	Conv	Kernel=3×3 FM=192 WN lReLU
8	Conv	Kernel=3×3 FM=192 WN lReLU
9	Conv	Kernel=3×3 Stride=4×2 FM=192 WN IReLU
10	Dropout	$p = 0.5$
11	Conv	Kernel=3×3 FM=192 WN IReLU
12	NiN	FM=192 WN IReLU
13	NiN	FM=192 WN IReLU
14	Pooling	FM=192 Global Pooling
15	Dense	FM=4 WN

B. Performance of Activity Recognition

We first illustrate the performance of DeepSeg on activity recognition. The distinctive component in DeepSeg is the CNN-based activity segmentation algorithm. Therefore, to explicitly reveal its segmentation effectiveness, we freeze the activity classification model and compare the performance of DeepSeg with the one when replacing the activity segmentation algorithm with other segmentation techniques. Since existing supervised segmentation methods do not exhibit superior performance compared with unsupervised methods [56] and cannot meet the requirement of the real-time nature of the applications needing activity segmentation, more recent studies focus on unsupervised methods for CPD-based activity segmentation [30], [57]. Therefore, we select a few unsupervised segmentation techniques as the baselines. For threshold-based approaches, we choose the three segmentation algorithms for CSI-based activity recognition [17], [14], [15]. For CPD-based approaches, we select the three representative methods [33], [36], [37], each of which can represent a kind of CPD approach. (1) **WiAG** [17]: A typical threshold-based segmentation method for gesture extraction. The method identifies the start and end of a gesture by comparing the amplitudes of the principal component streams with a given threshold. (2) **Tw-See** [14]: An effective activity segmentation method for the scenarios of the signals through the wall. This method can eliminate the influence of slight fluctuations of the waveforms and further accurately detect the start and end points of activities. (3) **Wi-Multi** [15]: A novel activity segmentation algorithm under noisy environment with multiple subjects. The algorithm can eliminate potential false detection by calculating the largest eigenvalues from both amplitude and calibrated phase correlation matrices, and can improve accuracy for noisy environments or/and multiple subject activities. (4) **AR1seg** [33]: A typical change point detection method in the area of statistics. This approach estimates change-points using an autoregressive process of order 1. (5) **SEPseg** [36]: A start-of-the-art method to detect change points for time series data. This algorithm uses separation distance as a divergence measure to detect change points and outperforms existing methods on real and synthetic data. (6) **CPDact** [37]: A CPD-based segmentation method specifically designed for activity recognition. This approach can efficiently enhance the performance of identifying activity boundaries and recognizing human daily activities. (7) **NoFeedback**: Our proposed DeepSeg but removing the feedback mechanism. This model uses the same segmentation algorithm and classification model as DeepSeg, but there is no feedback for the activity segmentation algorithm.

The results of activity recognition for mixed, coarse-grained and fine-grained activities are shown in Figure 5. For mixed activity, as shown in Figure 5(a), our proposed CNN-based segmentation algorithms, NoFeedback and DeepSeg, obviously outperform the threshold-based and CPD-based segmentation methods. This is primarily attributed to the deep learning model which can effectively deal with the difference between amplitudes of different kinds of activities. These two deep learning-based models can more accurately identify the start and end of activities. And therefore, despite using the same activity recognition model, both accurate and F1-score of their activity recognition are significantly higher than both threshold-based and CPD-based baselines. While, in these two CNN-based algorithms, DeepSeg further improves the recognition performance, which indicates that our proposed framework with the feedback mechanism can further enhance the final recognition performance.

In addition, for coarse-grained activities in Figure 5(b), since the amplitudes of their CSI data have relatively small differences, the threshold-based and CPD-based methods can accurately segment activities, and correspondingly acquire relatively high recognition accuracy. For example, the accuracy of WiAG arrives at near 94%, which is close to the result of NoFeedback. However, the performance of all the baselines is lower than our proposed DeepSeg. The similar trend can be found for fine-grained activities in Figure 5(c). These results suggest that, compared with the baseline methods, DeepSeg can efficiently improve the recognition performance even though there are only activities at similar levels.

C. Role of Activity Classification Models

The analyses in the previous section show that DeepSeg achieves better performance when using the CNN architecture as the activity classification model. To evaluate the effectiveness of the feedback mechanism in this framework, here we freeze the activity segmentation algorithm, and evaluate the performance of DeepSeg when replacing the CNN architecture with other machine learning models for activity classification. There are three typical approaches for CSI-based activity classification: Dynamic Time Warping (DTW), shallow learning and deep learning based methods [22], [15]. Since it is difficult for DTW-based methods to output classification probabilities of an input sample, which is required for DeepSeg, we select the support vector machine (SVM) and long short-term memory (LSTM) as the representations of shallow learning and deep learning based methods respectively. These two models are widely used for different CSI-based applications and achieve expected performance, such as gesture recognition [58], [41], fall detection [11], [47] and activity recognition [59], [60]. For the SVM model, we adopt the same features and settings as the work [11]. For the LSTM model, similar to [59], the model is composed of three LSTM hidden layers and one fully connected dense layer.

The accuracy and F1 score using the LSTM and SVM models are presented in Figure 7 and Figure 8, respectively, for mixed, coarse-grained and fine-grained activities. Here, DeepSeg-LSTM in Figure 7 means our proposed framework but replacing the CNN with the LSTM for activity classification, and NoFeedback-LSTM refers to the one without the feedback component. The similar explanation can apply to DeepSeg-SVM and NoFeedback-SVM in Figure 8. When using the LSTM for activity classification, shown in Figure 7, the performance of DeepSeg-LSTM is obviously better than that of NoFeedback-LSTM for different types of activities. For example, for mixed activity, the accuracy of DeepSeg-LSTM is around 3% higher than that of NoFeedback-LSTM. The similar trend can be found for the SVM in Figure 8. These results indicate that the framework with the feedback component always outperforms the ones without the feedback,

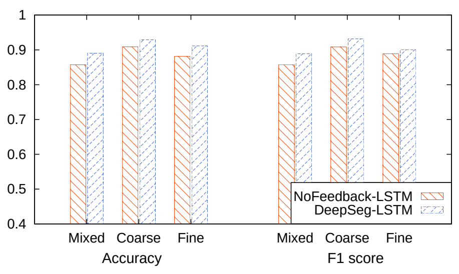


Fig. 7: The performance of activity recognition using LSTM

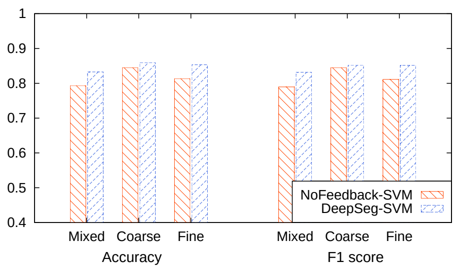


Fig. 8: The performance of activity recognition using SVM

and this framework can apply to different supervised learning models.

D. Performance of Activity Segmentation

We have shown that DeepSeg can consistently achieve better performance on activity recognition than the baseline methods. However, it is not clear how its activity segmentation performance is. Next, we conduct experiments to compare the segmentation accuracy of the activity segmentation algorithm from DeepSeg with three threshold-based methods (**WiAG** [17], **Tw-See** [14] and **Wi-Multi** [15]) and three CPD-based methods (**AR1seg** [33], **SEPseg** [36] and **CPDact** [37]). For a given activity, we define the segmentation accuracy as $|A \cap B|/\max\{|A|,|B|\}$, where A refers to the real set of packet indexes for this activity, and B is the predicted set. By intuition, accurate segmentation will be helpful to activity classification. Note that the activity classification accuracy may exceed the segmentation one, because an activity can be correctly classified by the activity classification model even though the start and end of this activity are not precisely detected. If the predicted start and end points are close to the real ones, the majority of the activity data can be extracted and might be adequate for activity classification.

Figure 9 presents the segmentation accuracy for coarse-grained, fine-grained and mixed activities. The solid lines represent the accuracies of the baseline methods with different thresholds. The dotted lines are the ones of DeepSeg. Since DeepSeg does not need the threshold, its accuracies are the horizontal lines. Note that Wi-Multi adopts a dynamic threshold to segment activities. However, for a given CSI series, the threshold is determined only by the sampling rate. The x-axis in Figure 9(c) refers to the sampling rate, and we call it the threshold for convenience of description. For AR1seg, a penalty is regarded as the threshold value, because it has the same function with the threshold. When a penalty is too small, many change points are detected, even those that are the result

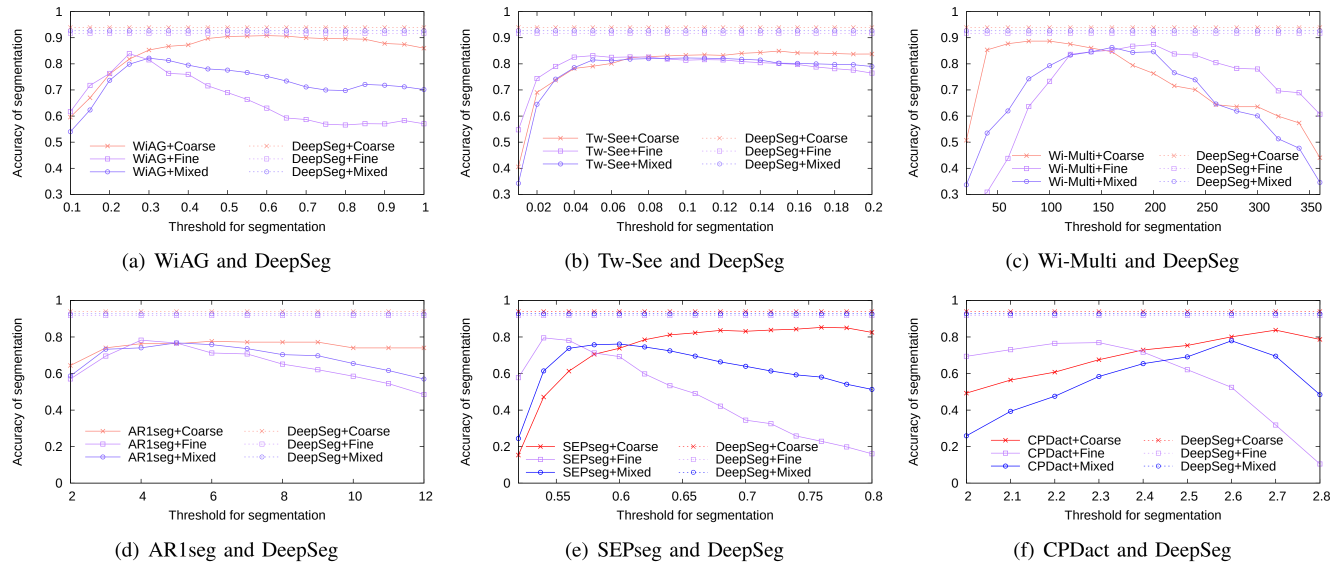


Fig. 9: The performance of activity segmentation. Here, Coarse, Fine and Mixed refer to coarse-grained activities, fine-grained activities and mixed activities respectively.

of noise. Conversely, too much penalization only detects the most significant changes, or even none [33], [29]. The x-axis in Figure 9(d) means the penalty, which is also called as the threshold.

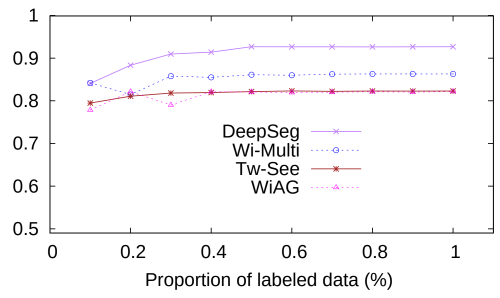
We make three observations from the figures. First, the segmentation performance of DeepSeg always outperforms the threshold-based and CPD-based methods no matter how the threshold changes, especially for mixed activities. For example, as shown in Figure 9, the accuracies of DeepSeg for mixed activities are around 10%, 10% and 6% higher than that of WiAG, Tw-See and Wi-Multi respectively. While, for activities with similar granularity, such as coarse-grained activities, the performance of the baseline methods tends to approach DeepSeg. This is because CSI amplitudes of these activities have a small difference, and further a given threshold can be used to accurately detect the start and end of these activities. Nevertheless, these six methods consistently exhibit an inferior performance compared to DeepSeg for different kinds of activities. These results strongly indicate that DeepSeg can effectively enhance segmentation performance compared with the baselines, especially for mixed activity.

Second, for DeepSeg, different types of activities obtain quite similar performance. For example, the difference between the accuracies of mixed activities and fine-grained activities is less than 1%. This suggests that our proposed DeepSeg can achieve relatively stable performance when applying to different kinds of activities. Third, for both threshold-based and CPD-based methods, activities with different granularity should adopt different thresholds to obtain better segmentation performance. For instance, for WiAG in Figure 9(a), when the threshold is 0.25, the segmentation accuracy of fine-grained activity arrives at the highest value, 83.89%. But coarse-grained activity only obtains 81.83% accuracy, which is near 9% lower than its best one at the threshold 0.6. The similar trend can be found for the other five baselines in Figure 9. These results suggest that these baseline methods can acquire good performance for activities with similar granularity. But when applying to mixed activities, they may suffer from distinct performance degradation. However, our proposed DeepSeg can obtain relatively stable expected performance for different kinds of activities.

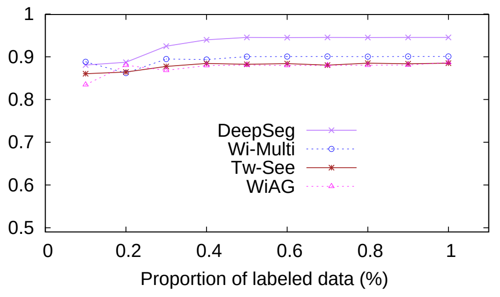
E. Impact of the Training Data Size

Here we exploit the impact of labeled data size on the segmentation and recognition performance. To this end, we keep the test data fixed, and select the different percent of training data as the training set, and the mixed activity data is used for these experiments. We compare our proposed DeepSeg with the three threshold-based methods: WiAG, Tw-See and Wi-Multi. For DeepSeg, the training set is used to train the state inference model which will be adopted to segment activities in the test set. For these three threshold-based methods, the training set is used to seek optimal thresholds for activity segmentation.

The segmentation accuracy and corresponding classification accuracy are shown in Figure 10(a) and Figure 10(b), individually. Figure 10(a) shows that DeepSeg obtains increasing segmentation accuracy with the rise of the labeled data size, which indicates that the labeled data size has an important impact on segmentation performance for our proposed method. Besides, except at a small number of training data such as 10% of data, DeepSeg significantly outperforms the three threshold-based methods. The reason behind this is because DeegSeg uses the deep learning model to segment activities and requires a certain number of samples for model training. However, when the data size reaches 20% of labeled data, the performance becomes obviously higher than the baselines. The similar trend can be found for the activity classification in Figure 10(b). These results indicate that DeepSeg requires a certain number of training data to obtain better performance. However, the number of labeled data, such as 20% of training

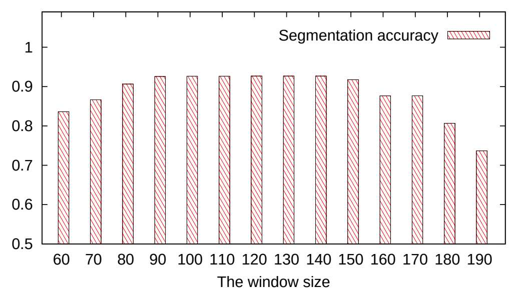


(a) The segmentation accuracy

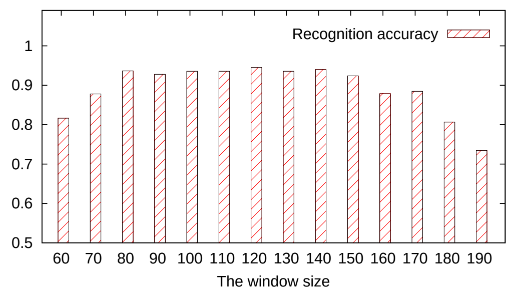


(b) The recognition accuracy

Fig. 10: The performance with different size of labeled data



(a) The segmentation accuracy



(b) The recognition accuracy

Fig. 11: The performance with different windows sizes

data which means 25 samples per category, is easily affordable by human labeling.

F. Impact of the Window Size

In the segmentation algorithm, we divide CSI streams into bins of equal size. In this set of experiments, we examine how the window size (the bin size) impacts the accuracy of activity segmentation and activity recognition. The mixed activity data is used for these experiments.

Figure 11(a) and Figure 11(b) present the accuracy of the activity segmentation and activity recognition, respectively, for various window sizes. The x-axis corresponds to the window size. The two figures show that the performance is decreasing when the bin size is becoming extremely small or large. The reason is that, when the size is too large, a whole activity data might fall into one bin. And, it is not easy to predict the state labels of this kind of bin by the state inference model, which makes it difficult to segment activities. When the size is too small, fluctuations caused by noises might be mistakenly regarded as activity data, which can degrades segmentation performance. In other words, DeepSeg can only apply to the scenario that there is an interval (such as more than two seconds) between two activities.

However, the accuracy of segmentation and recognition keeps relatively stable when the window size is near the middle part (ranging from 90 to 150). This indicates that our proposed algorithm can achieve expected performance for a wide range of the window sizes. In our data, the average and standard deviation of the length of activities are 219 and 58 individually. Hence, when the window size is close to the half of the average length of activities, DeepSeg can obtain stable and expected performance. These analyses indicate that the window size can influence the segmentation and recognition accuracy of DeepSeg. To obtain better performance, values near the half of the average length of activities can be selected as the window size.

V. Conclusions

In this paper, we presented DeepSeg, a deep learning-based activity segmentation framework for activity recognition using WiFi signals. Instead of adopting the threshold to segment activities, we proposed a CNN-based activity segmentation algorithm, which can avoid experience-dependent noise removal and threshold calculation and address the performance decline problem for mixed activities. Unlike existing approaches which treat activity segmentation and activity classification as two independent stages, we designed a feedback mechanism to refine the activity segmentation algorithm by considering the feedback extracted from activity classification results. Extensive experiments demonstrate that our proposed framework outperforms state-of-the-art methods for the scenario with fine-grained and/or coarse-grained activities.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] M. Z. Uddin and M. M. Hassan, “Activity recognition for cognitive assistance using body sensors data and deep convolutional neural network,” *IEEE Sensors Journal*, pp. 1–8, 2019.

[3] Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K. R. Choo, “Imaging and fusing time series for wearable sensor-based human activity recognition,” *Information Fusion*, vol. 53, pp. 80–87, 2020.

[4] J. Wannenburg and R. Malekian, “Physical activity recognition from smartphone accelerometer data for user context awareness sensing,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3142–3149, 2017.

[5] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, “A robust human activity recognition system using smartphone sensors and deep learning,” *Future Generation Computer Systems*, vol. 81, no. 4, pp. 307–313, 2018.

[6] R. Bodor, B. Jackson, N. Papanikolopoulos, and H. Tracking, "Vision-based human tracking and activity recognition," in *Proceedings of the 11th Mediterranean Conference on Control and Automation*, 2003, pp. 18–20.

[7] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recognition*, vol. 61, no. 1, pp. 295–308, 2017.

[8] H. Abdelnasser, K. A. Harras, and M. Youssef, "A ubiquitous wifi-based fine-grained gesture recognition system," *IEEE Transactions on Mobile Computing*, vol. 4, no. 11, pp. 1–14, 2018.

[9] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, 2018.

[10] H. Zou, Y. Zhou, R. Arghandeh, and C. J. Spanos, "Multiple kernel semi-representation learning with its application to device-free human activity recognition," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7670–7680, 2019.

[11] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "Falldefi: Ubiquitous fall detection using commodity wi-fi devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–25, 2018.

[12] L. Chen, X. Chen, L. Ni, Y. Peng, and D. Fang, "Human behavior recognition using wi-fi csi: Challenges and opportunities," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 112–117, 2017.

[13] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Computing Surveys*, vol. 52, no. 3, pp. 1–36, 2019.

[14] X. Wu, Z. Chu, P. Yang, C. Xiang, X. Zheng, and W. Huang, "Tw-see: Human activity recognition through the wall with commodity wi-fi devices," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 306–319, 2019.

[15] C. Feng, S. Arshad, S. Zhou, D. Cao, and Y. Liu, "Wi-multi: A three-phase system for multiple human activity recognition with commercial wifi devices," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7293–7304, 2019.

[16] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 65–76.

[17] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using wifi," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017, pp. 252–264.

[18] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[19] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 339–367, 2017.

[20] M. A. Wiering, H. van Hasselt, A. Pietersma, and L. Schomaker, "Reinforcement learning algorithms for solving classification problems," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2011, pp. 91–96.

[21] J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu, "Reinforcement learning for relation classification from noisy data," in *The Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 5779–5786.

[22] Z. Wang, B. Guo, Z. Yu, and X. Zhou, "Wi-fi csi-based behavior recognition: From signals and actions to activities," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 109–115, 2018.

[23] B. Sheng, F. Xiao, L. Sha, and L. Sun, "Deep spatial-temporal model based cross-scene action recognition using commodity wifi," *IEEE Internet of Things Journal*, pp. 1–10, 2020.

[24] Q. Bu, G. Yang, X. Ming, T. Zhang, J. Feng, and J. Zhang, "Deep transfer learning for gesture recognition with wifi signals," *Personal and Ubiquitous Computing*, no. 1, pp. 1617–4917, 2020.

[25] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: A real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2017.

[26] F. Wang, W. Gong, and J. Liu, "On spatial diversity in wifi-based human activity recognition: A deep learning based approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2035–2047, 2019.

[27] H. Yan, Y. Zhang, Y. Wang, and K. Xu, "Wiact: A passive wifi-based human activity recognition system," *IEEE Sensors Journal*, vol. 20, no. 1, pp. 296–305, 2020.

[28] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using wifi signals," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 363–373.

[29] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, pp. 1–20, 2020.

[30] S. Aminikhanghahi and D. J. Cook, "Enhancing activity recognition using cpd-based activity segmentation," *Pervasive and Mobile Computing*, vol. 53, pp. 75–89, 2019.

[31] Q. Ni, T. Patterson, I. Cleland, and C. Nugent, "Dynamic detection of window starting positions and its implementation within an activity recognition framework," *Journal of Biomedical Informatics*, vol. 62, pp. 171–x180, 2016.

[32] M. A. U. Alam, N. Roy, A. Gangopadhyay, and E. Galik, "A smart segmentation technique towards improved infrequent non-speech gestural activity recognition model," *Pervasive and Mobile Computing*, vol. 34, pp. 25 – 45, 2017, pervasive Computing for Gerontechnology.

[33] S. Chakar, E. Lebarbier, C. Lvy-Leduc, and S. Robin, "A robust approach for estimating change-points in the mean of an $\mathrm{AR}(1)$ process," *Bernoulli*, vol. 23, no. 2, pp. 1408–1447, 05 2017.

[34] M. Raja, V. Ghaderi, and S. Sigg, "Wibot! in-vehicle behaviour and gesture recognition using wireless network edge," in *IEEE 38th International Conference on Distributed Computing Systems*, 2018, pp. 376–387.

[35] M. Raja, Z. Vali, S. Palipana, D. G. Michelson, and S. Sigg, "3d head motion detection using millimeter-wave doppler radar," *IEEE Access*, vol. 8, pp. 32 321–32 331, 2020.

[36] S. Aminikhanghahi, T. Wang, and D. J. Cook, "Real-time change point detection with application to smart home time series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 1010–1023, 2019.

[37] S. Aminikhanghahi and D. J. Cook, "Enhancing activity recognition using cpd-based activity segmentation," *Pervasive and Mobile Computing*, vol. 53, pp. 75–89, 2019.

[38] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 90–102.

[39] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When csi meets public wifi: Inferring your mobile phone password via wifi signals," in *ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1068–1079.

[40] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with wi-fi!" *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2907–2920, 2016.

[41] J. Yang, H. Zou, Y. Zhou, and L. Xie, "Learning gestures from wifi: A siamese recurrent convolutional architecture," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 763–10 772, 2019.

[42] N. Yu, W. Wang, A. X. Liu, and L. Kong, "Qgesture: Quantifying gesture distance and direction with wifi signals," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, 2018.

[43] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 617–628.

[44] J. Yang, H. Zou, H. Jiang, and L. Xie, "Device-free occupant activity sensing using wifi-enabled iot devices for smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3991–4002, 2018.

[45] F. Xiao, J. Chen, X. H. Xie, L. Gui, J. L. Sun, and W. Ruchuan, "Seare: A system for exercise activity recognition and quality evaluation based on green sensing," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–10, 2018.

[46] L. Wenyuan, W. Siyang, and W. Lin, "A multiperson behavior feature generation model based on noise reduction using wifi," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 5179–5186, 2020.

[47] C. Han, K. Wu, Y. Wang, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," in *IEEE Conference on Computer Communications*, 2014, pp. 271–279.

[48] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, p. 115C124.

[49] H. Li, K. Ota, M. Dong, and M. Guo, "Learning human activities through wi-fi channel state information with multiple access points," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 124–129, 2018.

[50] C. Xiao, D. Han, Y. Ma, and Z. Qin, "Csigan: Robust channel state information-based activity recognition with gans," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 191–10 204, 2019.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.

[52] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.

[53] D. Zhao, Y. Chen, and L. Lv, "Deep reinforcement learning with visual attention for vehicle classification," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 4, pp. 356–367, 2017.

[54] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 2124–2133.

[55] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 53–53, 2011.

[56] K. D. Feuz, D. J. Cook, C. Rosasco, K. Robertson, and M. Schmitter-Edgecombe, "Automated detection of activity transitions for prompting," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 5, pp. 575–585, 2015.

[57] S. Deldari, D. V. Smith, A. Sadri, and F. Salim, "Espresso: Entropy and shape aware time-series segmentation for processing heterogeneous sensor data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, 2020.

[58] W. He, K. Wu, Y. Zou, and Z. Ming, "Wig: Wifi-based gesture recognition system," in *The 24th International Conference on Computer Communication and Networks*, 2015, pp. 1–7.

[59] C. Feng, S. Arshad, R. Yu, and Y. Liu, "Evaluation and improvement of activity detection systems with recurrent neural network," in *2018 IEEE International Conference on Communications*, 2018, pp. 1–6.

[60] L. Zhang, C. Wang, M. Ma, and D. Zhang, "Widigr: Direction-independent gait recognition system using commercial wi-fi devices," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1178–1191, 2020.

Chunjing Xiao received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China.

He is currently an Associate Professor with the School of Computer and Information Engineering, Henan University, Kaifeng, China. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA. His current research interests include online social networks, information retrieval, machine learning, wireless networks, and Internet of Things.

Yue Lei received the B.E degree from the School of Computer and Information Engineering, Henan University, China, in 2019. He is currently pursuing the M.S. degree in the School of Computer and Information Engineering, Henan University.

His research interests include deep learning, data analytics and signal processing.

Yongsen Ma received the B.S. degree in control science and engineering from Shandong University, Jinan, China, and the M.S. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China. He is currently pursuing the Ph.D. degree with the Department of Computer Science, College of William and Mary, Williamsburg, VA, USA.

He is currently a member of the LENS Research Group, Williamsburg, where he is advised by Dr. G. Zhou. He was a Research Assistant with Intel, Shanghai. His current research interests include wireless networking, ubiquitous sensing, and mobile systems.

Fan Zhou received the B.S. degree in computer science from Sichuan University, Chengdu, China, in 2003, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, in 2006 and 2012, respectively.

He is currently an Associate Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests include machine learning, spatiotemporal data management, and social network knowledge discovery.

Zhiguang Qin received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China.

He is a Full Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, where he is also the Director of the Key Laboratory of New Computer Application Technology and UESTC-IBM Technology Center. His current research interests include medical image processing, computer networking, information security, cryptography, information management, intelligent traffic, electronic commerce, distribution, and middleware.