A Multi-Task Deep Learning Approach for Sensor-based Human Activity Recognition and Segmentation

A Multi-Task Deep Learning Approach for Sensor-based Human Activity Recognition and Segmentation

Furong Duan, Tao Zhu, *Member, IEEE*, Jinqiang Wang, Liming Chen, *Senior Member, IEEE,* Huansheng Ning, *Senior Member, IEEE,* and Yaping Wan

Changes for Reviewer 1 highlighted in blue
Changes for Reviewer 4 highlighted in red

***Abstract*—Deep learning for sensor-based human activity recognition (HAR) has been a focus of research in recent years. Sensor data stream segmentation is a core element in HAR, which has currently been treated as an independent preprocessing task, usually with a fixed-size window. This has led to two critical problems, namely the multi-class window problem caused by possible multiple activities within a fixed-size window and the fluctuation of prediction results due to noisy data and over-segmentation. To address these research challenges, in this paper, we conceive a novel Multi-Task deep learning approach to segmenting and recognizing human activity simultaneously. Specifically, we propose a multi-scale window method based on feature sequence generation to overcome the multi-class window problem. We develop a novel boundary offset prediction algorithm to adjust a window's boundary to tackle the over-segmentation issue. In addition, We design a multi-task framework to streamline and optimize the activity recognition and segmentation tasks simultaneously. We conduct extensive experiments on eight benchmark datasets to evaluate the proposed framework and associated methods. Initial results show that our approach outperforms the performance of current state-of-the-art HAR methods.**

***Index Terms*—Deep learning, multi-task learning, activity recognition, activity segmentation, sensors**

I. INTRODUCTION

SENSOR-based Human activity recognition is a key technique in many real-world context-aware applications, such as smart homes and healthcare [1]–[3]. Early studies [4], [5] showed that although traditional approaches for HAR, such as ontological reasoning [6] and supportive vector machine [7] had achieved satisfactory results, they significantly relied on handcrafted feature engineering and domain knowledge from experts.

In contrast, deep learning (DL) can automatically extract low and high-level features by training an end-to-end neural network and has been very successful in image classification, video object detection, and natural language processing during the last decade. There has been growing interest in applying deep learning to HAR in recent years, leading to a large number of studies [8], [9].

Generally speaking, an activity recognition workflow includes the sensor data segmentation, feature extraction and activity classification. Many different models have been explored for feature extraction, including convolution neural network(CNN) [10], selective kernel convolution neural network [11], long short-term memory (LSTM) [12] and self-attention [13]. Nevertheless, Compared to feature extraction, human activity segmentation (HAS) has relatively attracted little attention.

In a real-world application scenario, sensor data streams need to be segmented in HAR systems in real-time. The challenge is it is difficult to define the activity boundaries [14]. Traditional HAS techniques using fixed-size windows have been proven to be ineffective due to the irregularity of the activity duration. As such, selecting a suitable window size is still a challenge. For example, jumping is a short-duration activity, and walking may last a long duration. A small window may lead to misclassification because it contains inadequate information. A large window could contain multiple activities providing more information. Traditional windowing approaches generally take the activity class of the largest number of occurrences as the label of the window [8], [14]. If the window size is not defined appropriately, there could have multiple classes in one window and these activity classes are not identical, leading to the so-called multi-class window problem [15]. The transitions and short periods of activity aggravated this problem, further reducing activity recognition performance.

In recent years, several approaches have been proposed to address the fixed-size window selection and multi-class window challenges. Noor et al. [16] and Akbari et al. [17] proposed an adaptive sliding window method that generates the default size window and then adjusts the size by comparing it with the activity class of the adjacent windows. Guédon [18], Keogh et al. [19], and Fryzlewicz et al. [20] presented change point detection methods. However, these methods are inappropriate in real-time continuous data streams because the boundaries of activities are uncertain. To tackle this problem, Yao et al. [15] exploited the idea of image semantic segmentation for sample-level prediction. This method usually results

This work is partly supported by the National Natural Science Foundation of China (62006110, 62071213).The Research Foundation of Education Bureau of Hunan Province (21C0311, 21B0424). Hengyang Science and Technology Major Project: 202250015428. (Corresponding author: Tao Zhu, Yaping Wan.)

Furong Duan, Tao Zhu, Jinqiang Wang, and Yaping Wan were with the Department of Computer Science, University of South China, 421001 China (e-mail: frduan@stu.usc.edu.cn, tzhu@usc.edu.cn, jqwang@stu.usc.edu.cn, ypwan@aliyun.com).

Liming Chen was with the School of Computing and Mathematics, University of Ulster, Belfast, BT37 0QB, U.K. Huansheng Ning was Department of Computer & Communication Engineering, University of Science and Technology Beijing, 100083 China (email: l.chen@ulster.ac.uk, ninghuansheng@ustb.edu.cn).

in some fluctuations in the predicted labels caused by noise or other factors' interference, leading to the so-called over-segmentation problem.

We hypothesize that jointly learning activity segmentation with recognition could be a novel approach to solving multi-class window and over-segmentation problems. Inspired by computer vision object detection, we propose a multi-scale window method to tackle the multi-class window problem. Unlike traditional windowing methods, we generate multi-scale windows based on feature sequences exploiting the receptive field features. To alleviate the over-segmentation problem, we propose a boundary offset prediction network to predict the activity boundary offset for adjusting the window length and a concatenated algorithm to get the final results of recognition and segmentation. In addition, considering that HAR and HAS share the same feature space, we propose a MTHARS(multi-task human activity recognition and segmentation) framework that can provide a mutual constraint on activity recognition and segmentation to streamline the process and optimize the performance of the proposed approach.

The main contributions of this article are as follows.

- We develop the MTHARS multi-task framework to streamline and optimize activity recognition and segmentation simultaneously, thus improving both tasks' performances.
- We propose a novel multi-scale window based segmentation method to address multi-class windows problem.
- For efficient segmentation, we design the activity boundary offset prediction network and a concatenated algorithm to tackle over-segmentation issue.
- We conduct comprehensive experiments on eight public activity datasets and evaluate the MTHARS method with the state-of-the-art results from relevant studies, demonstrating the effectiveness and superiority of the MTHARS method.

The rest of the article is organized as follows. Section II discusses related works. Section III details the proposed MTHARS approach and associated framework and methods. Section IV presents the experiments and results including discussions on research findings. Section V summarizes our work and discusses future work.

II. Related Works

A. Deep learning for Human Activity Recognition

Using DL for Sensor-based activity recognition has attracted increasing studies [10], [21]. A CNN for HAR usually consists of convolution, pooling, and a fully connected layers [22], [23]. Huang et al. [24] presented a shallower CNN to implement channel information interaction. To better capture the temporal and spatial features, recurrent neural network (RNN), like GRU [25] and LSTM [26], was introduced in HAR. Ordóñez et al. [27] proposed combining CNN and LSTM to fuse multimodal sensor information for improving activity recognition performance. Guan et al. [12] proposed to combine multiple LSTM learners into an ensemble classifier, making it more robust in solving realistic scenario challenges such as noisy, ambiguous data. More recently, attention mechanisms have been investigated for HAR models. Tang et al. [13] presented a ternary attention mechanism with channel, time, and sensor modalities attention. Khan et al. [28] assumed that generating multiple heads with attention can improve feature representations. Abedin et al. [29] introduced a self-attention encoder to learn the latent interactions between multiple sensor channels. Existing models that utilize CNNs for activity recognition typically have the same receptive field size in the neurons of the same layer, which presents a limitation to capture more features [11]. To address this, Gao et al. [11] proposed SK convolution, which uses the attention mechanism to learn multi-scale features by automatically modifying the receptive field.

These existing studies usually use fixed-size window methods for segmentation, suffering from the multi-class window problem. To alleviate this problem, Okeyo et al. [30] proposed a dynamic segmentation model that can shrink and expand the window size by analyzing sensor data, temporal activity information, and the status of the activity recognition. Noor et al. [16] proposed an adaptive sliding window method that compares the window with the activity class of the adjacent windows to adjust the window size. Akbari et al. [17] proposed a hierarchical signal segmentation method which first applied a larger window to extract features, and then divided the window into smaller windows based on the activity category probabilities and reassigned the labels. Yao et al. [15] developed a full convolution network that uses the dense labelling strategy to assign a class label to each data point. Triboan et al. [31] presented a semantic-based segmentation method inspired by ontological modelling.

B. Multi-task deep learning for Human Activity Recognition

Multi-task learning (MTL) is a learning paradigm, which is built upon the assumption that, for a set of related task, information contained for each task can help other tasks to improve model generalization. This is usually achieved by learning tasks simultaneously and sharing low-dimensional representation [32].

Ruder [33] summarized two MTL methods for deep learning: soft or hard parameter sharing of hidden layers. In soft parameter sharing, each task has its own model, but models have similar parameters. In hard parameter sharing, all tasks share the hidden layers among them. This can be further classified into three categories [34]: namely the multi-input single-output (MISO), the single-input multi-output (SIMO), and the multi-input multi-output (MIMO). It is commonly accepted that HAR and HAS are closely related in the way that both share the same feature space. Based on this observation, the information contained in one task could be helpful to the other, and an integrated approach to jointly performing two tasks could address the challenges associated with HAS and HAR separately in one go. Based on this classification scheme our framework in this paper belongs to SIMO. Zhang et al. [35] raised three research questions for MTL, i.e. when to share, what to share and how to share. According to the approach characterisation of [35], our method is a Feature Transformation Approach that falls into the category of feature-based.

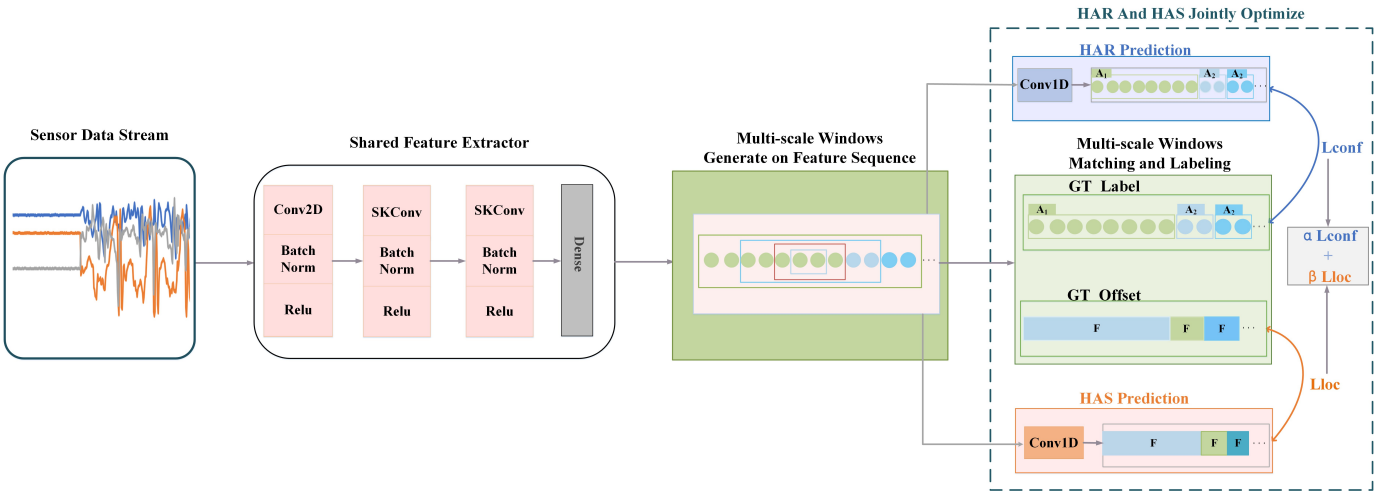


Fig. 1. Overview of the Proposed MTHARS framework (GT, Lconf, and Lloc represent Ground Truth, Classification Loss, and Offset Loss respectively.)

In the field of deep learning HAR, there are already studies on multi-task learning, but none of them address simultaneous HAR and HAS. Sun et al. [36], [37] proposed an online MTL framework that was for personalized human activity recognition, where each task corresponds to a specific person in activity recognition. Chen et al. [38] proposed a deep MTL approach, jointly solving activity recognition and user recognition. These studies have demonstrated the advantages of MTL. In this paper, we develop a novel MTL framework to optimize activity recognition and segmentation performance simultaneously.

III. The MTHARS Approach

In this section, we formally define the research problem and describe the MTHARS framework.

A. Problem definition

Given an activity data stream $\mathbf{X} = \{\mathbf{x_i}\}_{i=1}^{N}$, where $\mathbf{x_i}$ denotes a vector of signals collected at timestamp t. $\mathbf{a} = \{a_k\}_{k=1}^{K}$ is the set of K activity labels contained in the activity data stream, where $a_k \in \{A_1, A_2, A_3 \cdots A_L\}$, and each a_k has a different duration. Our goal is to segment the streaming sensor data into K non-overlapping segments $\{\mathbf{X}_k\}_{k=1}^{K}$, each $\mathbf{X}_k \subseteq \mathbf{X}$, and $\bigcup_{k=1}^{k=K} \mathbf{X}_k = \mathbf{X}$. Each $\mathbf{X}_k$ is assigned an a_k. The start and end of $\mathbf{X}_k$ represent an activity boundary, which can also be denoted by a centroid point and corresponding length (t^x, t^l).

B. The holistic framework of MTHARS

Traditional activity recognition methods usually use a fixed-size window to perform sensor data stream segmentation. This has led to the multi-class window and over-segmentation problems. To address them, we propose a multi-task framework to jointly optimize the activity recognition and segmentation tasks. The MTHARS framework consists of a selective kernel convolution neural network [11] (as a backbone network), multi-scale window generation module, and the HAR and HAS prediction module, as shown in Fig. 1. It works as follows: First, the backbone network generates the feature sequence. In order to solve the multi-class window problem, the window generation module produces a certain number of multi-scale windows. The HAR and HAS prediction module consists of two neural networks in parallel branches. One performs HAR tasks to predict the activity class contained in each window, and the other performs HAS tasks to predict the boundary offset between a window and the truth activity bounding box for addressing the over-segmentation problem. Our recognition and segmentation prediction flow, as shown in Fig.2, uses a non-maximum suppression(NMS) algorithm to retain the best windows. The final results are calculated by algorithm 1, which concatenates adjacent windows of the same activity.

C. Multi-scale windows generation on feature sequences

Representation of multiscale windows: Unlike the conventional sliding window used in the community of sensor-based HAR, we leverage the concept of anchor in the computer vision field [39]. A window is denoted by (x, l), where x is the window center, and l represents the window length.

Generation of multiscale windows: Assume that the provided feature sequence is n. With a scale of $s \in (0, 1]$, $n\sqrt{s}$ and $n/\sqrt{s}$ represent the window lengths. s takes the values of s_1, s_2 $\cdots$ s_m. This will produce $n \times m \times 2$ windows. Therefore, we can cover various truth activity lengths as much as possible.

We are inspired by the receptive field in convolutional operations to deal with the relationship between sensor data and feature sequences. So, we set the feature sequence length as half of an input data stream length. We generate windows centered on each unit of the feature sequence(as shown in Fig. 2) and divide the center of the window by the feature sequence length. Hence, the x indicates the window's relative position in the feature sequence. Since the centers of the window are distributed over all the feature sequence units, these centers are uniformly distributed over input data stream based on their relative spatial locations.

IOU: We employ the Jaccard index to measure the similarity between the generated window and the truth activity

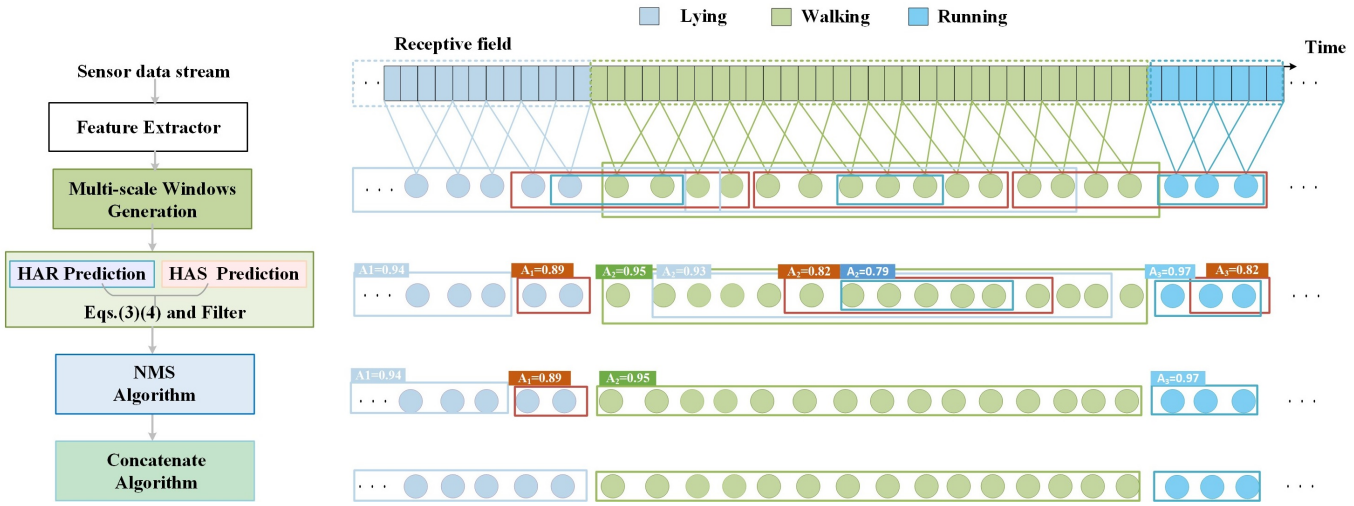


Fig. 2. Overview of the MTHARS's activity recognition and segmentation prediction flow. (A_1 = 0.98 represents a class probability of 0.98 for activity A_1 in this window.)

bounding box. The Jaccard index is defined as the ratio of the intersection length of the window and the truth activity bounding box to their merge length. We denote their Jaccard values as an intersection over union (IOU). The range of an IOU is between 0 and 1 that indicates whether there is any overlap. 0 indicates that there is no overlap, while 1 indicates that they are equal.

Multi-scale windows matching and labeling: Each generated window is treated as a training sample. To train the model, we need to label each window with an activity class and a bounding offset, where the former refers to the activity associated with the window and the latter refers to the offset of the truth activity bounding box concerning the window.

To mark any generated window, We apply the following approach to assign the truth activity bounding box to the window.

Assume the generated windows are W_1, W_2 $\cdots$, W_{na}, and the truth activity bounding boxes are T_1, T_2 $\cdots$, T_{nb}, where $na > nb$. First, we create a matrix $\mathbf{M} \in \mathbb{R}^{(n_a \times n_b)}$, in which the element m_{ij} in the i^{th} row and j^{th} column is the IOU of W_i and T_j. The steps for matching are as follows:

1) Find the largest element in **M**, we set its row index and column index to i_1 and j_1, respectively, and then assign T_{j_1} to W_{i_1}, discarding all the elements in the i_1 rows and j_1 columns of **M**.
2) Repeat the previous steps until all elements in column *nb* of **M** have been discarded. So that all of the truth activity bounding boxes are allocated to a generated window.
3) For each W_i of the remaining *na-nb* windows, find its best matching T_j, and assign T_j to W_i if the IOU is great than the threshold.

Here is a concrete example. Assume that the largest IOU in $\mathbf{M} \in \mathbb{R}^{(5\times3)}$ is m_{51}, T_1 is assigned to W_5. Then, we discard all the elements in Row 5 and Column 1 of **M**.

$$\mathbf{M} = \begin{bmatrix} 0.55 & 0.82 & \mathbf{0.96} \\ 0.69 & \mathbf{0.95} & 0.78 \\ 0.32 & 0.48 & 0.88 \\ 0.75 & 0.67 & 0.45 \\ \mathbf{0.98} & 0.67 & 0.88 \end{bmatrix}_{5\times3}$$

Next, we repeat the above steps to find the maximum m_{13}, m_{22} subsequently, and discard row 1 and column 3, row 2 and column 2. After that, we traverse through the remaining unassigned W_3, W_4 and assign them the truth activity bounding box based on their IOU against the threshold.

Now, we can label a window with an activity class and the boundary offset in the following way. If a window is assigned to a truth activity bounding box, the class of the truth activity bounding box is used to label that window. The boundary offset is calculated as follows.

Given a truth activity bounding box T= (t^x, t^l) and a window W = (w^x, w^l). We define the offset as F = (f^x, f^l), where f^x, f^l represents the centre and the length offset, respectively, calculated as follows:

$$f^x = \frac{t^x - w^x}{w^l} \tag{1}$$

$$f^l = \log \frac{t^l}{w^l} \tag{2}$$

Predicted activity boundary: In the predicting process, we employ the predicted offsets ($\hat{f}^x$, $\hat{f}^l$) to calculate the activity boundary ($\hat{t}^x$, $\hat{t}^l$), where $\hat{t}^x$, $\hat{t}^l$ denote the activity center and length, respectively.

$$\hat{t}^x = \hat{f}^x w^l + w^x \tag{3}$$

$$\hat{t}^l = w^l \exp(\hat{f}^l) \tag{4}$$

Non-maximum Suppression (NMS): For the same activity, it may be matched by multiple windows. To find the most suitable window, we modified the NMS algorithm from the computer vision field [40]. This algorithm is described as follows.

The HAR network predicts the probability p for each W. All the predicted windows are ranked as a list L based on p in descending order for the same activity. This list L is then processed as follows.

1) Select a W with the highest p as the base and remove the rest of the windows whose IOU with W over the threshold. Thus, W is retained, and the windows with high similarity to W are discarded. In other words, the window with the non-highest p is suppressed.
2) Repeat the above steps until all windows are selected or non-base windows discarded. As such, the IOU values of any two predicted windows are below the threshold.

Algorithm 1 Final Output

Input: Predicted activity-boundary offsets $\mathbf{F} = \{F_i\}_{i=1}^{N}$
Windows $\mathbf{W} = \{W_i\}_{i=1}^{N}$
Predicted activity classes $\{a_i\}_{i=1}^{N}$
Predicted activity probabilities $\{p_i\}_{i=1}^{N}$
Output: activity boundaries $\hat{\mathbf{T}} = \{\hat{T}_i\}_{i=1}^{K}$
activity classes $\hat{\mathbf{a}} = \{\hat{a}_i\}_{i=1}^{K}$

Remove redundant windows via the NMS algorithm
Calculate activity boundaries $\{\mathbf{X}_i\}_{i=1}^{L}$ with Eqs. (3)(4)
$a_1 \leftarrow$ The first activity of $\mathbf{X}_1$
$x_1 \leftarrow$ The starting position of the first activity in $\mathbf{X}_1$
$x_l \leftarrow$ The ending position of the first activity in $\mathbf{X}_1$
Initial index i
for $j = 1$ to L **do**
  if $a_1 \neq a_j$ **then**
    Add a_1 to $\hat{a}_i$
    Add (x_1, x_l) to $\hat{T}_i$
    $a_1 \leftarrow a_j$
    $x_1 \leftarrow x_l + 1$
    $i \leftarrow i + 1$
  end if
  $x_l \leftarrow$ boundary length of $\mathbf{X}_j + x_l$
end for
if $\hat{\mathbf{T}}$ is Empty **then**
  Add a_1 to $\hat{a}_i$
  Add (x_1, x_l) to $\hat{T}_i$
end if

D. HAR and HAS Prediction

This component is comprised of two neural networks in parallel branches for HAS and HAR tasks. For the HAR branch, suppose the number of activity classes is k, and each generated window has k+1 classes, where class 0 represents the background. The length of the feature sequence is n. When centered on each unit of the feature sequence generating m windows, a set of $n \times$ m $\times$ 2 windows need to be classified. Choosing a convolutional layer can reduce the number of parameters and does not change the length of the feature sequence. Hence, the output and input coordinates correspond to each other. To generate effective predictions, there are $m(k+1)$ output class channels. For the same spatial position, the output channel with the index $i(k+1)+j$ $(0 \leq j \leq k)$ denotes the predictions of class j for the window i.

The design of the HAS branch is similar to the HAR branch. The only difference is that we predict two offsets for each window. We utilize the predicted offset and the absolute position of the window to accurately locate the boundary of the activity. Hence, we are able to address the over-segmentation problem and segment the activity more accurately.

E. Joint Losses

The boundary offset loss (5) is calculated by comparing the truth offset $F = (f^x, f^l)$ with the predicted offset $\hat{F} = (\hat{f}^x, \hat{f}^l)$ using Smooth_{L_1} loss.

$$L_{loc}(F, \hat{F}) = \sum_{i \in \{x,l\}} Smooth_{L1}(F_i - \hat{F}_i) \tag{5}$$

in which

$$Smooth_{L1}(x) = \begin{cases} 0.5(x)^2, if|x| < 1. \\ |x| - 0.5, otherwise. \end{cases} \tag{6}$$

The classification loss is calculated by the window labeled activity class $\mathbf{a}$ with the predicted class $\hat{\mathbf{a}}$ using cross-entropy loss (7), in which n represents the sample number.

$$L_{conf}(\mathbf{a}, \hat{\mathbf{a}}) = -\sum_{i=1}^{n} a_i log(\hat{a}_i) \tag{7}$$

Some windows not matched with any activity are regarded as negative samples. We sort all the negative samples in descending order according to the activity class probability and select a certain number of negative samples with a higher class probability to participate in the classification loss. The number of negative to positive samples is in the ratio of 3:1. If all the negative samples are involved in the training, it will make the training process tend towards the negative samples.

Finally, we multiply the weight α by the classification loss (conf) plus the weight β by the localization loss (loc).

$$L(a, \hat{a}, F, \hat{F}) = \frac{1}{N}(\alpha L_{conf}(a, \hat{a}) + \beta L_{loc}(F, \hat{F})). \tag{8}$$

F. Activity recognition and segmentation prediction

Our prediction process is shown in Fig.2. After the multi-scale windows generation module, these windows are put into the trained HAR and HAS networks to predict the window categories and the offsets. Next, the length of the windows was adjusted by (3)(4). Windows with low class probability are filtered out. Then we employ a NMS algorithm to remove similar windows. As the length of an activity input to the network is fixed, we input a fixed length activity data stream at a time to recognize the boundary of each activity. Finally, we concatenate all the segments according to the activity class to obtain the boundary of each activity in the whole activity data stream. The concatenation algorithm[1] is described in the algorithm 1.

[1]https://github.com/duanfurong/multi-scale-windows

TABLE I
SIMPLE DESCRIPTION OF PUBLIC HAR DATASETS.

Attribute \ Dataset	SKODA	HCI	PS	WISDM	UCI	OPPORTUNITY	PAMAP2	UNIMIB SHAR
Type	AG	AG	ADL	ADL	ADL	ADL	ADL	ADL
Subject	1	1	4	29	30	4	9	4
Rate	96HZ	96HZ	50HZ	20HZ	50HZ	30HZ	33.3HZ	30HZ
Window Size	1s	1s	2s	10s	2.56s	1s	1s	1s
Activity Categories	10	8	6	6	6	18	18	17
Sample	696975	7352	161959	1,098,208	748406	701366	2872533	11771
Proportion of Training Data	70%	70%	70%	70%	70%	70%	80%	70%
Proportion of Testing Data	30%	30%	30%	30%	30%	30%	20%	30%

[1] AG denotes gesture activity and ADL denotes activity of daily life.

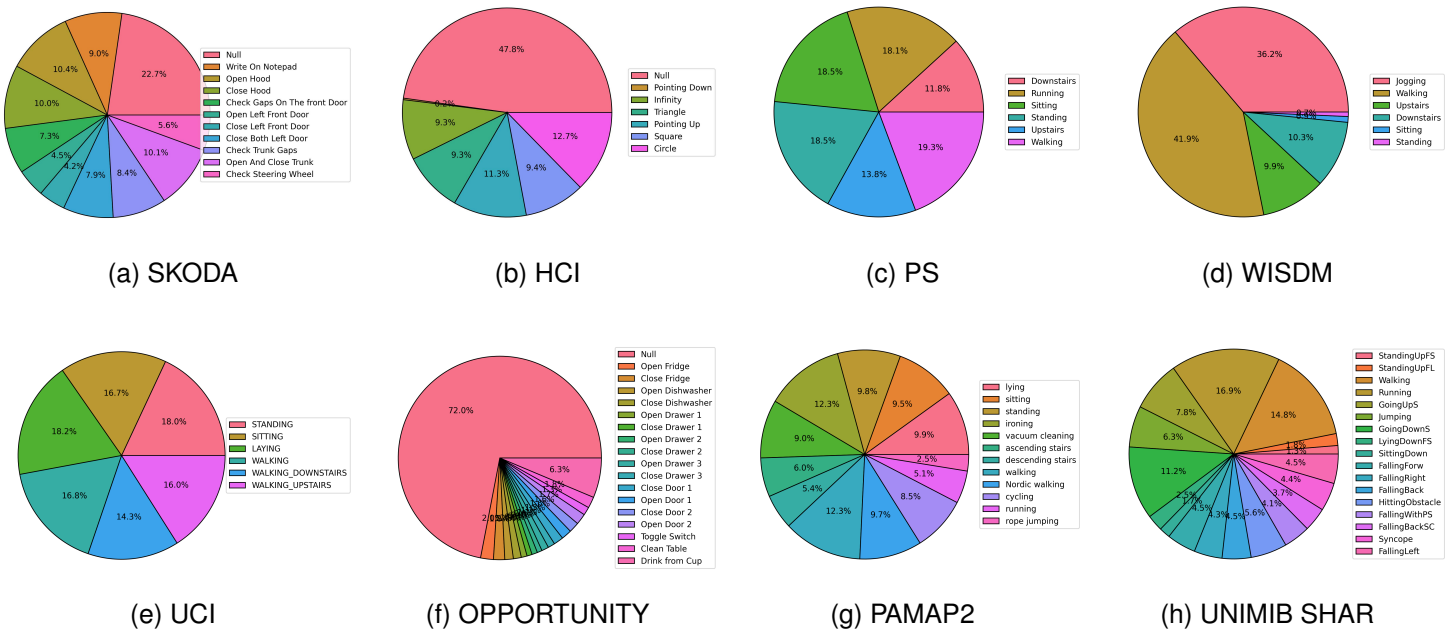


Fig. 3. Activity length distribution of benchmarking datasets

IV. EXPERIMENTS AND RESULTS

In this section, we detail our extensive experiments on eight benchmark datasets and analyze the experimental results to evaluate the effectiveness and scalability of the proposed MTHARS framework.

A. Datasets

We use eight publicly available benchmark HAR datasets to assess the effectiveness of our framework in daily activity segmentation and recognition. To make a fair comparison with other studies, we select the same parameters. The specifics of the eight benchmark datasets are described in Table I, and the activity length distribution are depicted in Fig. 3.

- SKODA Dataset [41]: A participant wearing 10 USB acceleration sensors in the left and right hands, respectively, performed 10 different gestural activities in an automobile repair scenario. Each of the gesture activities was conducted more than 70 times.
- HCI Dataset [42]: Eight accelerometer USB sensors were worn on the participant's right arm to perform various gestures, such as sketching triangles, squares, and circles.
- PS Dataset [43]: Four participants recorded their walking, standing, running, sitting, going upstairs, and going downstairs activities. They employed the phone's built-in accelerometer, magnetometer, and gyroscope on different locations: pants pocket, waistband, right arm and right wrist.
- WISDM Dataset [44]: The data were obtained by 29 participants using phones with triaxial acceleration sensors placed in their pant pockets with a sampling frequency of 20 Hz. Walking, strolling, walking up stairs, walking down stairs, standing motionless, and standing up were among the six daily activities undertaken by each participant. The mean value of the column was used to fill in the missing values in the dataset.
- UCI Dataset [45]: The dataset were generated by 30 participants aged 19 to 48 years performing daily activities.

TABLE II
THE OVERALL VERIFICATION TASKS.

Sections	Compared Baselines	Tasks
Section V.C	Dynp [18], BottomUP [19], and BinaryCPD [20]	activity segmentation
Section V.D	SK [11], Other methods in [12], [13], [15], [21], [23], [25], [26], [27], [28], [29], [47], [48]	activity recognition

Each participant wore a Samsung Galaxy S2 smartphone around their waist. They collected sensor data in 9 dimensions using the phone's built-in acceleration, gyroscope, linear acceleration, and 3-axis angular velocity sensors. Six different daily activities (walking, walking upstairs, walking downstairs, standing still, standing, and lying down) were performed.

- OPPORTUNITY Dataset [45]: IMU sensors were placed on volunteers' 12 different positions. They were asked to repeat a sequence of 17-morning activities in the kitchen, such as opening the refrigerator door, closing the refrigerator door, opening the drawer, closing the drawer, and so on. Interpolation was performed to fill in missing values in the dataset.
- PAMAP2 Dataset [46]: Nine participants wore an IMU on their chest, hands, and ankles, collecting three types of sensor information. Each participant was required to complete 12 mandatory activities, such as lying, standing, and walking up and down stairs, as well as six elective activities, such as watching television, driving, and playing ball. During the activity transformation, interference was generated. The start and the last 10 s of each activity were eliminated to reduce the noise data.
- UNIMIB SHAR Dataset [49]: The dataset was collected from the University of Milano-Bicocca. 30 volunteers equipped with Bosh BMA220 sensor Samsung cell phones. These were placed in their left or right pockets to collect daily life activities (running, going upstairs, standing, walking, and so on.) and different falling activities. Each activity was repeated 3 to 6 times.

B. Experimental Setup

We implement the MTHARS framework in the Pytorch platform and train it in Ubuntu 20.04 environment with an RTX 3090 GPU. We set the training epoch as 500 and use the Adam optimizer with an initial learning rate of 0.001. All datasets are divided into training and testing sets, and all the results are performed on the testing set.

1) *Evaluation metric:* To evaluate the accuracy of the activity segmentation, we use the Normalized Edit Distance (NED) [50]. NED uses the Levenshtein distance to measure the distance between the predicted activity sequence ($\hat{T}$) and the truth activity sequence (T), calculated by the minor operation that makes the two sequences equivalent. NED is defined as follows.

$$NED = \frac{lev(\hat{T}, T)}{length\,of\,T} \tag{9}$$

$$lev(i,j) = \begin{cases} \max(i,j)\,\min(i,j) = 0 \\ min = \begin{cases} lev(i-1,j)+1 \\ lev(i,j-1)+1 \\ lev(i-1,j-1)+1_{i\neq j}. \end{cases} \end{cases} \tag{10}$$

Equation (10) is the smallest operation step that makes the predicted sequence and the truth sequence equal and takes into account three different ways to make the two sequences equal, namely, removing an element from the sequence, inserting a new one, and changing the sequence's label directly.

Since the activity classes in human activity data are mostly unbalanced, using classification accuracy is not an appropriate criterion [27]. Therefore, we apply F_1 to evaluate the activity recognition performance.

$$F_1 = 2\sum \frac{N_c}{N_{total}} \frac{P \times R}{P + R} \tag{11}$$

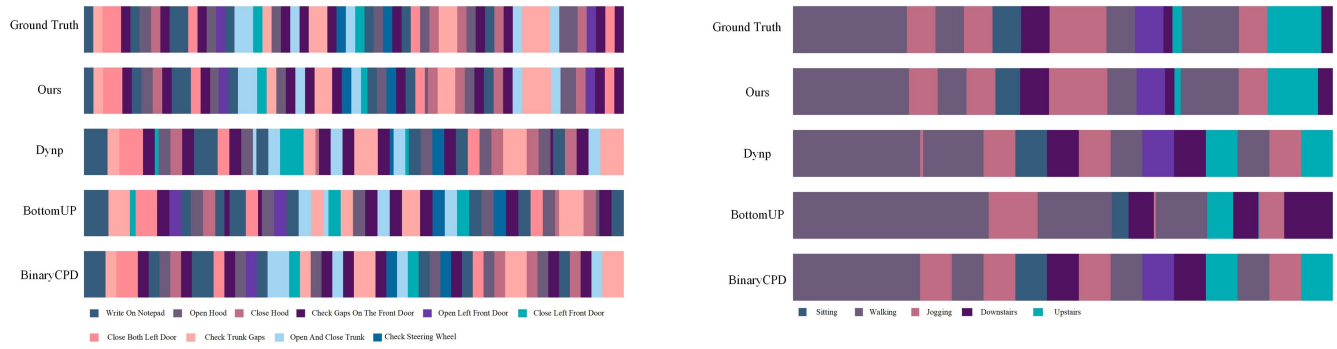
Here, N_c denotes the number of class c in all samples, and N_{total} the number of all samples. *P* and *R* are calculated from the set of all positive classes, defined below.

$$P = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \tag{12}$$

$$R = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \tag{13}$$

Here, i denotes the activity class in the dataset, TP_i the true positive class, FP_i the false-positive class and FN_i the false-positive class.

2) *Overall verification tasks:* We compare various tasks in the MTHARS framework with corresponding tasks in previous studies. Table II describes the verification tasks of MTHARS, including activity segmentation and recognition, respectively.



(a) HAS result on the SKODA(samples 1500-2500)

(b) HAS result on the WISDM(samples 3000-3400)

Fig. 4. HAS visualization results on SKODA and WISDM datasets

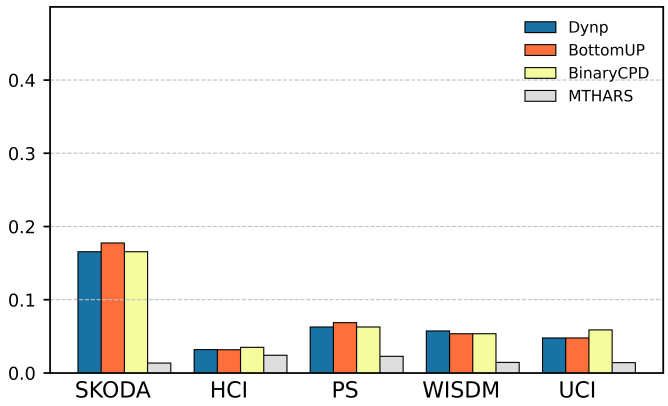


Fig. 5. NED performance on the five benchmark datasets

C. *Comparisons with dynamic segmentation algorithms*

In order to choose optimal window lengths for HAR and HAS, we adopt dynamic segmentation approaches, aiming to investigate if MTHARS can achieve better segmentation results with the help of classification tasks. Table III displays the segmentation results.

TABLE III
F_1 VALUE OF DYNAMIC SEGMENTATION.

Methods \ Datasets	SKODA	HCI	PS	WISDM	UCI
Dynp [18]	0.8858	0.8751	0.9663	0.8711	0.9352
BottomUP [19]	0.8661	0.8750	0.9536	0.8807	0.9353
BinaryCPD [20]	0.8826	0.8450	0.9601	0.8859	0.9151
MTHARS	**0.9648**	**0.9479**	**0.9733**	**0.9872**	**0.9632**

As revealed in Table III, all methods achieve over 90% F_1 values on PS and UCI datasets, though the performance of all approaches on PS is better than that on UCI. The MTHARS framework is superior to other dynamic segmentation approaches. One possible reason is that MTHARS, with the help of the activity recognition task, can improve the segmentation performance.

On the HCI dataset, The NED values of Dynp and BottomUP are similar and their classification results are similar. On the PS dataset, the NED values of Dynp and BinaryCPD are lower than the BottomUP method, and the F_1 values of their classifications are higher than the BottomUP classification. In addition, on the WISDM dataset, the BottomUP and BinaryCPD approaches have higher classification results than Dynp, and their NED values are lower than those of the Dynp method. Especially, We find out that the NED performance on the SKODA dataset is poor due to the fact that the SKODA dataset was collected in an unrestricted environment where each activity lasts for an irregular duration and the number of ground truth segments is relatively small. The degradation caused by this problem is noticeable for sequences with short-time or transition activities. These short-duration activities introduce errors in the testing phase, making it harder for the HAR models to further enhance the recognition performance.

As can be seen in Fig. 5, MTHARS achieves the lowest NED values. At the same time, the activity recognition is more accurate. The reason is that although some activities are short duration, in most cases, they are moderate length, which can reduce the difficulty of activity recognition. Our activity classification results are relatively higher, and our NED values are more stable. This result demonstrates that MTHARS, with the help of the activity recognition task, can improve the performance of segmentation.

To better depict the segmentation results, we visualize the segmentation performance in Fig.4. We observe that the SKODA dataset contains a lot of short-time activities leading to multi-class window problems. For the WISDM dataset, where most of the activities are long duration activities, baseline methods yield over-segmentation problems for short-duration activities. These can be alleviated by using a multi-scale window and offset prediction method.

From the above results, we observe that accurate activity segmentation can improve activity recognition performance. In addition, if data segmentation is considered a preprocessing process, errors in data segmentation may be propagated to the later steps. We can combine activity recognition and segmentation to facilitate each other.

TABLE IV
F_1 PERFORMANCE ON EIGHT DATA SETS.

Methods \ Dataset	SKODA	HCI	PS	WISDM	UCI	OPPORTUNITY	PAMAP2	UNIMIB SHAR
SK [11]	0.9510	0.9377	0.9574	0.9725	0.9558	0.9074	0.9338	0.7463
MTHARS	**0.9632**	**0.9524**	**0.9721**	**0.9877**	**0.9723**	**0.9213**	**0.9480**	**0.7571**
Other Reachers	0.924 [12]			0.949 [21]	0.9660 [13]	0.726 [12]	0.854 [12]	0.7538 [13]
	0.916 [15]			0.9263 [26]	0.9293 [23]	0.849 [21]	0.9248 [13]	
	0.958 [27]			0.9720 [28]	0.9302 [47]	0.8058 [48]	0.9116 [23]	
	0.928 [29]				0.9585 [26]	0.915 [27]	0.9303 [48]	
					0.9545 [25]	0.9263 [26]	0.908 [29]	
					0.9537 [28]	0.746 [29]		

TABLE V
ACCURACY& FLOPS(M) PERFORMANCE ON EIGHT DATASETS.

Methods \ Dataset	SKODA	HCI	PS	WISDM	UCI	OPPORTUNITY	PAMAP2	UNIMIB SHAR
SK [11]	0.9586&5.43	0.9341&4.34	0.9649&14.03	0.9751& 17.63	0.9406&8.95	0.9014&10.3	0.9380&45.38	0.7589&6.51
MTHARS	**0.9648&4.74**	**0.9479&3.79**	**0.9733&12.79**	**0.9872&6.62**	**0.9632&8.15**	**0.9153&9.0**	**0.9450&38.02**	**0.7648&5.01**

D. *Comparisons with activity recognition algorithms*

In this section, Our purpose is to verify that MTHARS can boost the performance of activity recognition accuracy with the help of segmentation tasks. Table IV and Table V detail the F_1 value and accuracy recognition performance, respectively.

Performance Gains: From Table IV, V, MTHARS exceeds the SK [11] performance with lower model complexity, regardless of which dataset it is based on. For example, on the SKODA dataset, compared with the SK performance, our framework obtains a 0.22 % higher F_1 value. Regarding classification accuracy, our framework surpasses the SK by 0.62 % with lower model complexity. Compared with SK, MTHARS obtains a 1.47 % higher F_1 value on the HCI dataset. On the PS dataset, MTHARS achieves 1.47 % F_1 value performance gains with little increase in the computational burden. These comparisons indicate that the features learned by activity recognition and segmentation contain unique information for HAR and can facilitate each other, leading to better performance.

For the WISDM dataset, previous research has found out that it is difficult to separate the activities of "DownStairs" and "UpStairs" from Fig.6, as they have similar patterns. Our proposed framework improves the classification results of both activities. This indicates that the activity classification performance can be better improved with the help of the activity segmentation task. Take the UCI dataset as an example, among all the activities, the best classification result is the "lying" due to the distinct orientation. The activities "sitting" and "standing" have very similar patterns. Therefore, the classification results of both activities are relatively low. MTHARS achieves the best performance among these six activities. The reason is that our framework considers both activity class features and activity boundary features. As shown in Fig.7, we plot the confusion matrices of MTHARS and SK on the PAMAP2 dataset. It can be observed that our framework improves the performance of the activities of "sitting" and "standing". These results confirm the advantages of our framework in terms of recognition performance.

In addition, we also compare MTHARS with some state-of-the-art approaches on the HAR datasets, and experimental results demonstrate in Table IV. A detailed description of these methods can be found in section II. It can be concluded that our proposed approach can achieve superior performance.

SK first proposed to adopt the concept of attention to conduct kernel selection among multiple branches with different receptive fields to recognize activity accurately. Compared with SK, we are a multi-task learning framework. Regarding the scope of the application, we can more accurately recognize activity and activity boundaries, and we have lower model complexity. We conclude that the MTHARS framework, performing activity segmentation and recognition simultaneously, is able to improve activity recognition performance. The activity segmentation employing multi-scale windows and offset prediction can provide an accurate boundary for activity recognition. The two tasks are therefore complementary.

E. *Ablation experiments*

In this subsection, we aim to investigate the effectiveness of our framework. We find the class loss L_{conf} and offset loss L_{loc} weights, and the scale s of the window are two essential settings.

1) Impact of Loss Formulation: Loss formulation is presented in Section III, where L_{conf} and L_{loc} represent the

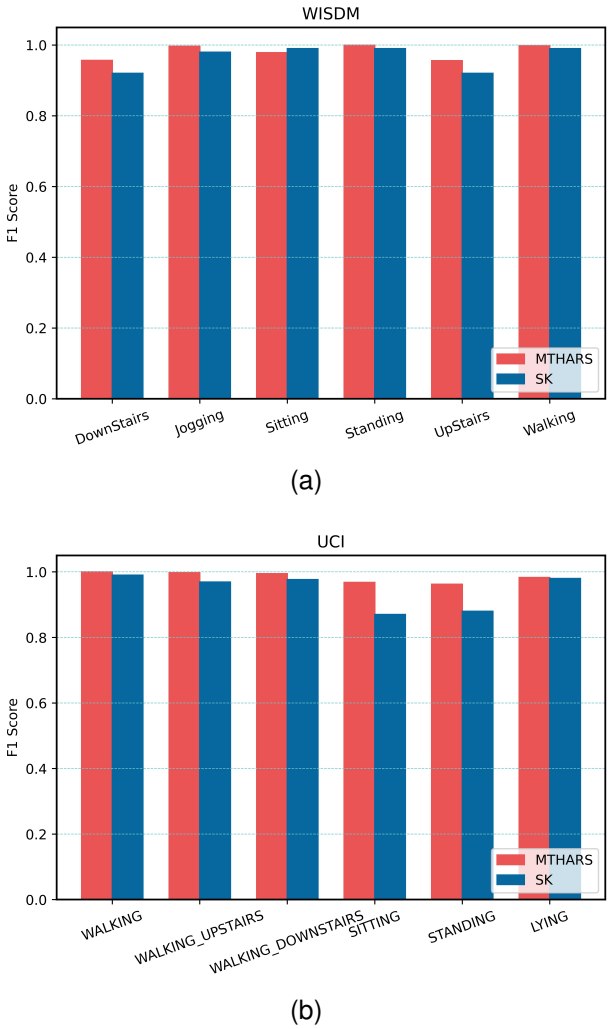


Fig. 6. Comparison of recognition results (F_1 value) for specific classes of human activities

classification loss(7), and boundary offset loss(5), respectively. For simplicity, the scale s of the window is fixed to 2 and 3. The detailed results of the experiment are shown in Table VI. We observe that the best result on the WISDM dataset is $2L_{conf}$ + $3L_{loc}$. In addition, the best result is very similar to the result of L_{conf} + L_{loc}. The results between L_{conf} + $2L_{loc}$ and $2L_{conf}$ + L_{loc} are close. These can be attributed to the weight of L_{conf} and L_{loc} being alike. Similarly, on the OPPORTUNITY dataset, we find the performance of L_{conf} + $3L_{loc}$ is close to $2L_{conf}$ + $3L_{loc}$. Moreover, The L_{conf} + L_{loc} brings a 1.53 % performance gain compared with the L_{conf} + $2L_{loc}$ on the OPPORTUNITY dataset. Furthermore, we compare the L_{conf} + L_{loc} with SK and achieve a noticeable improvement. In summary, We conclude that activity recognition and segmentation can be mutually facilitated. The components of our loss function can promote activity recognition performance.

2) Impact of scale s: Different lengths and numbers of scale s cause different effects. Therefore, we set various combinations of scales s to investigate its performance. Specifically, we set the loss combination as L_{conf} + L_{loc}. Due to the dataset length limitation, we only set the following settings, and the

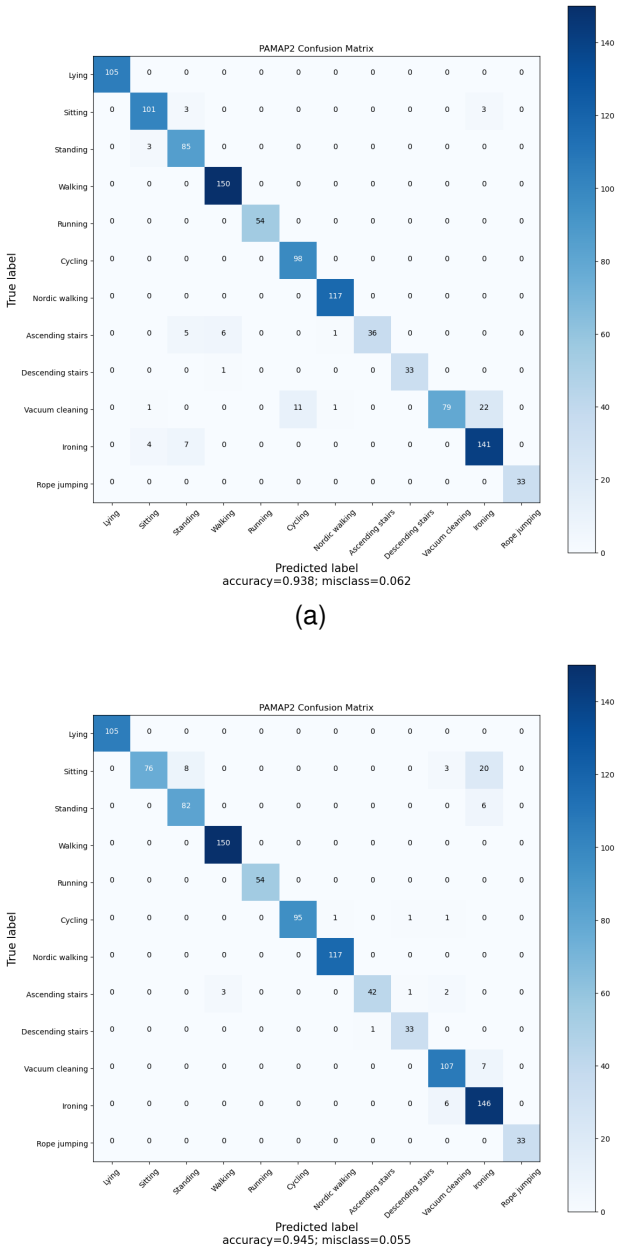


Fig. 7. The confusion matrices on the PAMAP2 dataset between SK [11] and MTHARS. (a) SK, (b) MTHARS

TABLE VI
ACTIVITY CLASSIFICATION F_1 VALUE WITH DIFFERENT WEIGHT SETTINGS ON BENCHMARK DATA SETS

Model	OPPORTUNITY	WISDM
SK [11]	0.9074	0.9725
L_{conf} + L_{loc}	**0.9213**	0.9877
L_{conf} + $2L_{loc}$	0.9060	0.9796
L_{conf} + $3L_{loc}$	0.9174	0.9874
$2L_{conf}$ + L_{loc}	0.9075	0.9783
$2L_{conf}$ + $3L_{loc}$	0.9154	**0.9881**
$3L_{conf}$ + L_{loc}	0.9111	0.9841

detailed results are shown in Table VII. When only set to a s, we observe that the performance is inferior to that of several different s. The reason is that the duration of activity is variable, and using a single-size window is insufficient to capture the activity characteristics of different durations. Similarly, setting a small scale on the OPPORTUNITY dataset, such as s = 0.5 and 0.3, may suffer from the same issue. The best classification performance is to set s = 2, 3. As most activity duration is irregular, this combination can better enhance activity recognition performance. At the same time, on the UCI dataset, the best result is 0.9723 when set s=2, 3, 4. The second result is 0.9615 with s=2, 3. One possible reason is that most activities are long duration, and therefore a larger window is able to collect more features. We conclude that combining classification and segmentation using multi-scale windows can better capture enough activity information to boost the HAR performance.

TABLE VII
ACTIVITY CLASSIFICATION F_1 VALUE FOR DIFFERENT s ON BENCHMARK DATASET

Model	OPPORTUNITY	UCI
SK [11]	0.9074	0.9558
s=2	0.9138	0.9505
s=0.5, 0.3	0.9160	0.8928
s=2, 3	**0.9213**	0.9615
s=2, 3, 4	0.9167	**0.9723**

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a multi-task deep learning framework for sensor-based human activity recognition and segmentation called MTHARS to jointly improve segmentation and recognition performance. We have developed a multi-scale windows method to address the multi-class window problem. We have also designed an activity boundary offset prediction network and a concatenated algorithm to tackle the over-segmentation issue. We have conducted comprehensive experiments on a number of publicly available datasets and analyzed the performance of HAR and HAS with the MTHARS against community recognized state-of-the-art methods. The initial results have demonstrated the effectiveness of the MTHARS. In the future, we plan to address issues related to model complexity so that this new integrated HAR and HAS framework can be deployed on mobile devices and used in real time.

TABLE VIII
MAIN ABBREVIATIONS AND SYMBOLS DESCRIPTION

Abbreviations or Symbols	Meaning
DL	deep learning
HAR	human activity recognition
HAS	human activity segmentation
NED	normalized edit distance
IOU	ratio of the intersection
NMS	non-maximum suppression
L_{conf}, L_{loc}	classification loss, localization loss
x, l	data point, data length
p	a class probability
a	predicted activity
A_i	activity class; $a \in \{A_1, A_2 \cdots\}$
s	window length scale
$W = (w^x, w^l)$	window value
$T = (t^x, t^l)$	truth boundary value
$F = (f^x, f^l)$	boundary offset value
$\mathbf{M}$	IOU matrix
α, β	classification, location weights

REFERENCES

[1] D. J. Cook and N. C. Krishnan, *Activity learning: Discovering, recognizing, and predicting human behavior from sensor data*. Wiley, 2015.

[2] A. Patel and J. Shah, "Sensor-Based Activity Recognition in the Context of Ambient Assisted Living Systems: A Review," *J. Ambient Intell. Smart Environ.*, vol. 11, no. 4, pp. 301–322, 2019. [Online]. Available: https://doi.org/10.3233/AIS-190529

[3] Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Systems with Applications*, vol. 137, pp. 167–190, 2019.

[4] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[5] O. D. Lara and M. a. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6365160

[6] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974, 2011.

[7] O. Brdiczka, J. L. Crowley, and P. Reignier, "Learning situation models in a smart home," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 56–63, 2009.

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

[9] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–40, 2021.

[10] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[11] W. Gao, L. Zhang, W. Huang, F. Min, J. He, and A. Song, "Deep Neural Networks for Sensor-Based Human Activity Recognition Using Selective Kernel Convolution," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

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

[13] Y. Tang, L. Zhang, Q. Teng, F. Min, and A. Song, "Triple Cross-Domain Attention on Human Activity Recognition Using Wearable Sensors," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022.

[14] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Activity recognition with evolving data streams: A review," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–43, 2018.

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

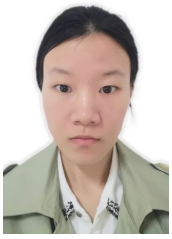
[16] M. H. M. Noor, Z. Salcic, K. K. I. Wang, I. Kevin, and K. K. I. Wang, "Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer," *Pervasive and Mobile Computing*, vol. 38, pp. 41–59, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.pmcj.2016.09.009

[17] A. Akbari, J. Wu, R. Grimsley, and R. Jafari, "Hierarchical signal segmentation and classification for accurate activity recognition," *UbiComp/ISWC 2018 - Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pp. 1596–1605, 2018.

[18] Y. Guédon, "Exploring the latent segmentation space for the assessment of multiple change-point models," *Computational Statistics*, vol. 28, no. 6, pp. 2641–2678, 2013.

[19] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm

for segmenting time series," in *Proceedings 2001 IEEE international conference on data mining*. IEEE, 2001, pp. 289–296.
[20] P. Fryzlewicz, "Wild binary segmentation for multiple change-point detection," *The Annals of Statistics*, vol. 42, no. 6, pp. 2243–2281, 2014.
[21] A. A. Varamin, E. Abbasnejad, Q. Shi, D. C. Ranasinghe, and H. Rezatofighi, "Deep auto-set: A deep auto-encoder-set network for activity recognition using wearables," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2018, pp. 246–253.
[22] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8553–8562, 2019.
[23] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.
[24] W. Huang, L. Zhang, W. Gao, F. Min, and J. He, "Shallow convolutional neural networks for human activity recognition using wearable sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.
[25] L. Tong, H. Ma, Q. Lin, J. He, and L. Peng, "A Novel Deep Learning Bi-GRU-I Model for Real-Time Human Activity Recognition Using Inertial Sensors," *IEEE Sensors Journal*, 2022.
[26] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56 855–56 866, 2020.
[27] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
[28] Z. N. Khan and J. Ahmad, "Attention induced multi-head convolutional neural network for human activity recognition," *Applied Soft Computing*, vol. 110, 2021.
[29] A. Abedin, M. Ehsanpour, Q. Shi, H. Rezatofighi, and D. C. Ranasinghe, "Attend and Discriminate: Beyond the State-of-the-Art for Human Activity Recognition Using Wearable Sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–22, 2021.
[30] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," *Pervasive and Mobile Computing*, vol. 10, pp. 155–172, 2014.
[31] D. Triboan, L. Chen, F. Chen, and Z. Wang, "A semantics-based approach to sensor data segmentation in real-time activity recognition," *Future Generation Computer Systems*, vol. 93, pp. 224–236, 2019.
[32] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
[33] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," no. May, jun 2017. [Online]. Available: http://arxiv.org/abs/1706.05098
[34] K.-h. Thung and C.-y. Wee, "A brief review on multi-task learning," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29 705–29 725, nov 2018. [Online]. Available: https://doi.org/10.1007/s11042-018-6463-xhttp://link.springer.com/10.1007/s11042-018-6463-x
[35] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[36] X. Sun, H. Kashima, R. Tomioka, N. Ueda, and P. Li, "A new multi-task learning method for personalized activity recognition," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2011, pp. 1218–1223.
[37] X. Sun, H. Kashima, and N. Ueda, "Large-scale personalized human activity recognition using online multitask learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2551–2563, 2013.
[38] L. Chen, Y. Zhang, and L. Peng, "METIER: A deep multi-task learning based activity and user recognition model using wearable sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–18, 2020.
[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
[40] A. Neubeck and L. V. Gool, "Efficient Non-Maximum Suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006.
[41] T. Stiefmeier, D. Roggen, and G. Troster, "Fusion of string-matched templates for continuous activity recognition," in *2007 11th IEEE International Symposium on Wearable Computers*. IEEE, 2007, pp. 41–44.
[42] K. Forster, D. Roggen, and G. Troster, "Unsupervised classifier self-calibration through repeated context occurences: Is there robustness against sensor displacement to gain?" in *2009 international symposium on wearable computers*. IEEE, 2009, pp. 77–84.
[43] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014.
[44] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
[45] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, 2013, pp. 437–442.
[46] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th international symposium on wearable computers*. IEEE, 2012, pp. 108–109.
[47] F. Cruciani, A. Vafeiadis, C. Nugent, I. Cleland, P. McCullagh, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, "Feature learning for human activity recognition using convolutional neural networks," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 18–32, 2020.
[48] 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 Journal*, vol. 20, no. 13, pp. 7265–7274, 2020.
[49] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.
[50] S. Aminikhanghahi and D. J. Cook, "Enhancing activity recognition using CPD-based activity segmentation," *Pervasive and Mobile Computing*, vol. 53, pp. 75–89, 2019. [Online]. Available: https://doi.org/10.1016/j.pmcj.2019.01.004



Furong Duan received his B.E. degree from Hengyang Normal University in 2019. Her is currently a M.S. student in the School of Computer Science, University of South China. His research interests include intelligent perception and pattern recognition.

Tao Zhu received his B.E. degree from Central South University, Changsha, China, and Ph.D. from University of Science and Technology of China, Hefei, China, in 2009 and 2015 respectively. He is currently an associate professor at University of South China, Hengyang, China. He is the principal investigator of several projects funded by the National Natural Science Foundation of China and Science Foundation of Hunan Province etc. He is now the Chair of IEEE CIS Smart World Technical Committee Task Force on "User-Centred Smart Systems".His research interests include IoT, pervasive computing, assisted living and evolutionary computation.

Jinqiang Wang received his B.E. degree from Henan Normal University in 2020. He is currently a M.S. student in the School of Computer Science, University of South China. His research interests include intelligent perception and pattern recognition.

Liming Chen is a Professor of Data Analytics in the School of Computing, Ulster University, UK. He received his BEng and MEng degrees at Beijing Institute of Technology, China, and DPhil on Computer Science at De Montfort University, UK. His current research interests include pervasive computing, data analytics, artificial intelligence and user centered intelligent systems and their applications in health care and cybersecurity. He has published over 250 papers in the aforementioned areas. Liming is an IET Fellow and a Senior Member of IEEE.

Huansheng Ning received his B.S. degree from Anhui University in 1996 and his Ph.D. degree from Beihang University in 2001. He is currently a Professor and Vice Dean with the School of Computer and Communication Engineering, University of Science and Technology Beijing and China and Beijing Engineering Research Center for Cyberspace Data Analysis and Applications, China, and the founder and principal at the Cybermatics and Cyberspace International Science and Technology Cooperation Base. He has authored several books and over 70 papers in journals and at international conferences/workshops. He has been the Associate Editor of the IEEE Systems Journal and IEEE Internet of Things Journal, Chairman (2012) and Executive Chairman (2013) of the program committee at the IEEE International Internet of Things Conference, and the Co-Executive Chairman of the 2013 International Cyber Technology Conference and the 2015 Smart World Congress. His awards include the IEEE Computer Society Meritorious Service Award and the IEEE Computer Society Golden Core Member Award. His current research interests include the Internet of Things, Cyber Physical Social Systems, electromagnetic sensing and computing.

Yaping Wan received his B.S. degree from Huazhong University of Science&Technology(HUST) in 2004 and his Ph.D. degree from HUST in 2009. He is currently a Professor and Dean with the School of Computer, University of South China and the International Cooperation Research Center for Medical Big Data of Hunan Province. He has authored several books and over 40 papers in journals and at international conferences/workshops. He has been the Workshop Chairman (2022) at the 16th IEEE International Conference on Big Data Science and Engineering, and the Session Chairman (2021,2022) of Asian Conference on Artificial Intelligence Technology. His current research interests include intelligent nuclear security, big data analysis and causal inference, high-reliability computing and security evaluation.