Metrol. Meas. Syst., Vol. 32 (2025), No. 1
DOI: 10.24425/mms.2025.152778

METROLOGY AND MEASUREMENT SYSTEMS
Index 330930, ISSN 0860-8229
www.metrology.wat.edu.pl

A MULTI-SCALE TIME SEGMENTS ATTENTION MECHANISM FOR SENSOR-BASED HUMAN ACTIVITY RECOGNITION

Hailong Rong, Hao Wang, Xiaohui Wu, Tianlei Jin, Ling Zou

Changzhou University, Changzhou 213164, China (✉ *rhle_16@163.com*)

Abstract

Sensor-based Human Activity Recognition (SHAR) technology is dedicated to utilizing sensor signals from smart devices to detect and identify human activities, thereby assisting in daily life. With the successful application of deep learning techniques, researchers are exploring the potential of integrating them with SHAR. Traditional fixed sliding window methods for processing datasets often lead to multi-class activity mixing. To alleviate this issue, researchers have introduced time attention mechanisms to focus on key temporal points related to activities. To Addressing this challenge, we propose an innovative Multi-scale Time Segments Attention Mechanism (MTSA), which diverges from traditional time attention mechanisms by focusing on time segments pertinent to activities, better aligning with the characteristics of SHAR data and significantly reducing computational resource consumption. Our experiments on recognized datasets such as UCI-HAR, PAMAP2, and WISDM validate the effectiveness of MTSA, demonstrating that it can be seamlessly integrated into existing SHAR models, enhancing performance without adding extra computational overhead.

Keywords: Time series classification, human activity recognition, Attention Mechanism, deep learning.

1. Introduction

With the increasing popularity of smartphones and other wearable devices such as smart bands and smart glasses, *Human Activity Recognition* (HAR) has become a very popular research topic. At the same time, HAR has very important applications in various fields, especially in scenarios requiring real-time monitoring such as healthcare [1], sports training [2], and smart homes [3]. Currently, HAR primarily relies on two methods: one is vision-based [4]; the other is sensor-based, mainly using inertial sensors attached to the human body. In recent years, researchers have also proposed new HAR methods based on WIFI channel state information [5], LiDAR [6], and millimeter-wave radar [7]. However, these and vision-based HAR methods often suffer from high costs, limited usability, sensitivity to environmental obstructions, and potential privacy concerns. In contrast, *Sensor-based Human Activity Recognition* (SHAR) offers a higher cost-performance ratio and has more distinct advantages in terms of comfort and privacy protection [8].

The standard process of SHAR includes five steps: data collection, data preprocessing, window segmentation, feature extraction, and action category output. Early SHAR often employed *machine learning* (ML) methods for feature extraction, which typically relied on shallow, handcrafted features such as mean, variance, amplitude, and frequency statistics [9–11]. While ML performs well in recognizing low-level activities such as standing and walking, it often falls short in identifying complex activities. In recent years, researchers have found that SHAR methods based on *deep learning* (DL) significantly outperform traditional machine learning approaches in context-aware and fine-grained activities [12].

Article history: received May 30, 2024; revised September 2, 2024; accepted October 9, 2024; available online February 9, 2025.

In the field of DL, common methods for feature extraction mainly include *Convolutional Neural Networks* (CNNs) and *Recurrent Neural Networks* (RNNs), as well as various variants of them. Although CNNs were originally designed for image processing, they have also shown some advantages in time series classification tasks. Teng *et al.* [13] introduced an efficient dynamic network called RepHAR, which utilizes structural reparameterization techniques to decouple the model during training and inference processes. Yang *et al.* [14] proposed an optimal activity graph generation model that converts time series into images, which are then recognized using convolutional neural networks. Gao *et al.* [15] used a multi-branch CNN structure that can adaptively select between branches with different convolutional kernels, effectively obtaining a dynamic receptive field. RNNs are specifically designed to process sequential data, maintaining connections with previous information through hidden states and continuously updating with new inputs. The most common variants of RNNs are *Long Short-Term Memory* (LSTM) and *Gated Recurrent Unit* (GRU). Ishimaru *et al.* [16] proposed using a bidirectional LSTM (BiLSTM) to recognize and record users' reading activities, with BiLSTM combining forward and backward LSTMs to more comprehensively capture contextual information in the data. Combining the application of CNNs and RNNs has become a mainstream methodology. This fusion strategy optimizes the model's ability to capture spatiotemporal features, thus significantly improving recognition accuracy when dealing with spatial data with temporal dependencies. Yao *et al.* [17] combined CNN and GRU modules in their proposed DeepSense framework, effectively avoiding noise interference in sensors and optimizing energy consumption and latency, as GRUs are more efficient than LSTMs. Zhang *et al.* [18] used an improved ResBiLSTM network to extract time series features from data processed by 1DCNN. The ResBiLSTM network integrates residual structures and layer normalization into a BiLSTM. This integration enhances the network's feature extraction capabilities.

Furthermore, attention mechanisms have been widely applied in the field of DL-based SHAR. Attention mechanisms dynamically adjust feature responses, enhancing the model's ability to capture key information, thereby achieving better results in complex activity recognition tasks. Ding *et al.* [19] utilized the *Squeeze-and-Excitation* (SE) module to obtain the weight relationships between tensor channels after convolution, and used the weighted data as input for the LSTM. The SE compresses the spatial information of feature maps into a channel descriptor through global average pooling, then learns the inter-channel dependencies using two fully connected layers, and obtains the weights for each channel through a sigmoid function [20]. Jitpattanakul and Mekruksavanich [21] introduced the *Efficient Channel Attention* (ECA) mechanism to avoid creating complex attention components, which uses a single-layer 1D convolution to replace the two fully connected layers in SE, with the aim of reducing computational resource consumption.

Despite the substantial progress made by SHAR methods based on DL in feature extraction, there are still challenges. First, determining the optimal window size for activity coverage remains an unresolved issue. In practical applications, due to the uncertainty of different activity cycles, using a fixed-size window to cover individual complete activities is impractical [21]. To address this issue, many scholars have introduced temporal attention mechanisms, which, unlike the channel attention mechanisms such as SE and ECA mentioned earlier, are techniques for processing sequential data. It allows the model to assign different levels of importance or attention to information at different time steps when processing temporal data. For example, the *Convolutional Block Attention Module* (CBAM) [22] and *Self Attention* (SA) mechanism [23] can capture attention features in the temporal dimension of time series data. CBAM performs calculations for channel and spatial attention, where spatial attention can be understood as temporal attention in time series data. Finally, the feature maps obtained from these two attention mechanisms are merged to enhance the network's feature representation capability. SA first initializes the Query, Key, and Value matrices of the input sequence, then calculates the similarity scores between the Query and all Keys. Then, the scores are converted into a probability distribution using the softmax function, and finally, the Value matrix is weighted and summed according to this probability distribution to obtain the final output vector. SE was originally designed to compute channel attention, but by swapping the channel and time dimensions in the data, temporal attention can also be calculated. Lu and Deng [24] applied the masking concept from image restoration to temporal signal processing, simplifying the U-Net used for denoising in images and incorporating residual structures and CBAM. This is to focus on the targets to be identified in the data. Mekruksavanich *et al.* [25] first used CNN layers to analyze sensor data and extract spatial features in their research. Then, these features were used to provide temporal sequence context for the BiLSTM network. Finally, through the CBAM attention mechanism, the model can focus on the most critical information in the BiLSTM feature maps. Wang *et al.* [26] proposed a new deep multifeature extraction framework based on attention mechanisms (DMEFAM). This mechanism incorporates self-attention mechanisms and CBAM, while also integrating CNNs and *bidirectional gated recurrent units* (Bi-GRU). Therefore, DMEFAM can extract a rich variety of features.

However, existing methods often allocate attention to every time point within a window, overlooking the fact that activities typically occur in time segments. This approach not only leads to unnecessary waste of computational resources but also, due to the uneven distribution of attention between adjacent time points, may disrupt the intrinsic dependencies within time series data. Zheng [27] proposed a model called LGSTNet, which divides the activity window into multiple sub-windows and allocates attention to each sub-window. This method alleviates the aforementioned issues to some extent, but due to the fixed size of sub-windows, it may lead to problems of over-segmentation or multi-class windows [21].

To address the limitations inherent in existing methodologies, particularly the elevated computational demands of attention mechanisms that can impose significant performance constraints in resource-limited settings, and acknowledging that many contemporary attention frameworks are primarily designed for visual tasks and thus may not constitute the most efficacious approach for HAR, this study proposes a new attention mechanism: *Multi-Scale Time Segments Attention* (MTSA). Unlike the coarse calculation of attention for each timestamp or fixed-size time segments, MTSA divides the time series at different scales and applies attention weighting on this basis. This mechanism not only considers the overall features of the time series but also captures key information at different time scales, thereby enhancing the model's ability to represent time series data.

The structure of this paper is as follows: Chapter 2 introduces the proposed MTSA. Chapter 3 describes the experimental environment and evaluation metrics. Chapter 4 presents the experimental results. Chapter 5 provides conclusions.

2. Method

2.1. Task Description

In this study, our goal is to identify the user's current activity from data collected by *Inertial Measurement Units* (IMUs). An IMU typically incorporates a 3-axis accelerometer and a 3-axis gyroscope to measure acceleration and angular velocity in three-dimensional space, known as a 6-axis IMU. Nowadays, IMUs can include a magnetometer in addition to the accelerometer and gyroscope, making it a 9-axis IMU. We assume that the user wears m IMUs, each containing n sensors, thus each sampling point includes N sensor readings, where $N = m \times n \times 3$. In SHAR, the prediction target is often a period of data to identify the activity occurring during that time. Therefore, a fixed-size sliding window is often used to segment the original sequence.

Specifically, setting the window size to *W* and the step size to *S*, the sliding window operation can be represented as:

$$\{x_t, x_{t+1}, \ldots, x_{t+W-1}\}, \{x_{t+S}, x_{t+S+1}, \ldots, x_{t+S+W-1}\}, \ldots \quad (1)$$

where x_t represents the value of the time series at time *t*. The step size *S* is generally less than *W*, ensuring that all data from the original sequence is included and the number of samples is increased. The size of the window is often determined by the sensor's sampling rate, with the optimal choice being able to cover the duration of a complete action within the activity.

The set of activity labels can be represented as $Y = \{1, 2, 3, \ldots, k\}$, where *k* denotes the number of types of activities. Therefore, in the dataset, each sample can be represented as $S_i = (w_i, y_i)$, $w_i \in \mathbb{R}^{N\times W}$, $y_i \in Y$. Specifically, our task is to predict the label y_i from the data of a window w_i. Fig. 1 illustrates the framework of a SHAR system. In this diagram, the structure of the feature extraction component is presented as the baseline model architecture used in the experimental section.

The baseline model proposed in this study consists of five consecutive basic blocks, each comprising a convolutional layer, a *batch normalization* (BN) layer, a *Rectified Linear Unit* (ReLU) activation layer, and a MTSA layer. The model concludes with processing via a *multilayer perceptron* (MLP), which ultimately yields the classification results.

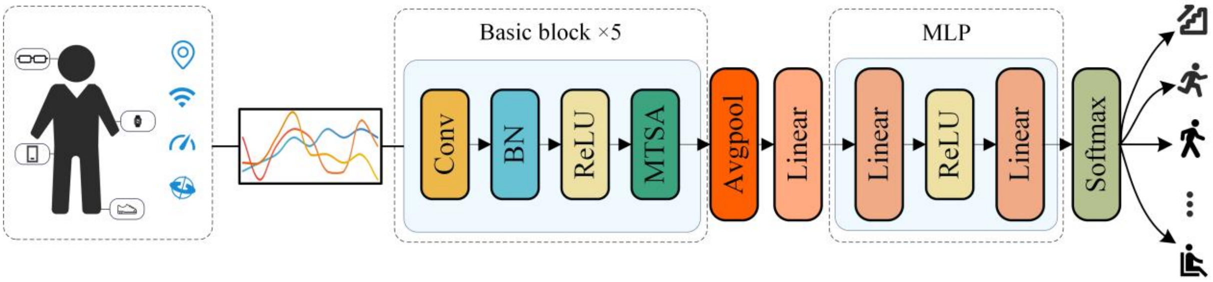


Fig. 1. Overview of the SHAR framework and proposed baseline model.

2.2. Multi-Scale Time Segments Attention

From the task description above, we find that using a fixed-size window to segment the dataset may pose a problem, as the data within the window might contain multiple activities. This heterogeneity of data could negatively impact the model's predictive accuracy. To overcome this challenge and enhance the key information in the data features, we propose a novel attention mechanism for HAR data, MTSA, which focuses on weighted processing of features over time periods rather than features of a single point. The details of this mechanism will be described below.

In the processing of sequential data, whether it is CNNs or RNNs, the extracted feature vectors inherently possess a temporal attribute. Features are extracted corresponding to the order of elements' appearance in the original sequence. Consequently, features related to the target activity often cluster together, appearing in the form of time segments. To capture the time segments of event-related features, the MTSA algorithm initially divides each input feature vector into B equal segments. Within these segments, at least one or more contain the occurrence of the event, while other segments may include data of non-target events. Such processing aims to isolate the event itself from the data and subsequently increase the weight of this data portion. The original input data $F \in \mathbb{R}^{C\times L}$, after being grouped, yields $F' \in \mathbb{R}^{B\times C\times L/B}$, where *C* represents the number of data channels, and *L* represents the length of the data. To explicitly model the dependencies between different time segment subsets, we use a time segment statistical information to describe the global information of each time segment.

This statistical information is described using the mean and maximum values of all data within this time segment. Studies have shown that the combination of these two types of data can effectively improve the model's accuracy [22]. The calculation steps for obtaining the statistical information z_b of the *b*-th time segment feature are as follows:

$$z_b = \{\frac{1}{C \times L / B} \sum_{i=1}^{C} \sum_{j=1}^{L/B} F_b'(i, j); \max_{1 \le i \le C, 1 \le j \le L/B} F_b'(i, j)\}. \tag{2}$$

To obtain $Z \in \mathbb{R}^{2\times B}$, all statistical information from the *B* time segments is integrated. In order to utilize the statistical information of these time segments to capture the relevant dependencies between them, we employ a layer of one-dimensional convolution to adaptively fuse this statistical information. Previous research often used a scaling factor r to reduce computational costs [20, 22], but an incorrect setting of r could lead to a certain degree of information loss. Our method compresses the time segment data, achieving a significant reduction in data volume, allowing a single-layer convolution to complete the fusion, thus avoiding the risk of overfitting due to excessive training parameters. The calculation method for the generated time segment weight attention $A \in \mathbb{R}^{1\times B}$ is as follows:

$$A = \sigma(\mathrm{Conv1d}(Z)), \tag{3}$$

where Conv1d denotes a 1-D convolutional layer, with the kernel size set to 3, and both stride and pooling size set to 1. σ represents the sigmoid function. The final output is obtained by multiplying the elements of attention A with each element in the corresponding time segment, and concatenating all time segments data along the first dimension. The calculation steps are as follows:

$$\tilde{F} = F' \times A, \tag{4}$$

$$\tilde{F} \in \mathbb{R}^{B\times C\times(L/B)} \xrightarrow{\text{Reshape}} \tilde{F}' \in \mathbb{R}^{C\times L}. \tag{5}$$

To accurately capture the temporal granularity of activity occurrences, we divide the input feature vectors along the time dimension into multiple time segments of varying scales and compute their time block attention. Finally, we fuse the weighted data across different scales. This approach allows MTSA to adaptively capture the time granularity required for different activities. The specific calculation steps are illustrated in Fig. 2.

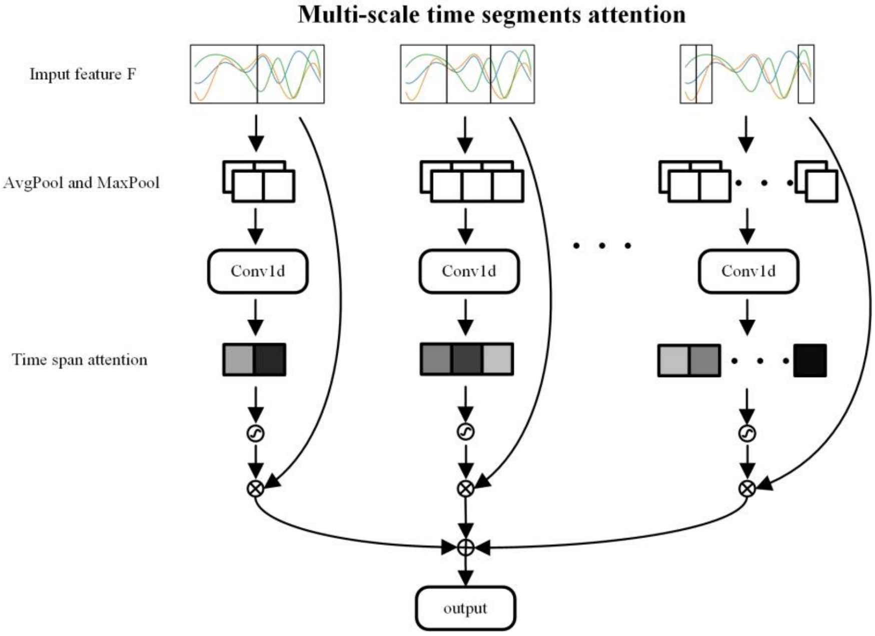


Fig. 2. Detailed calculation steps for MTSA.

3. Experimental setup

3.1. Dataset introduction

To conduct a comprehensive evaluation of our model, we consider multiple factors, including sampling rate, sensor type and quantity, as well as categories of activities. Drawing on these dimensions, we select three datasets as benchmarks to ensure the effectiveness and reliability of our model. The UCI-HAR and WISDM datasets mainly cover simple activities in everyday life, whereas PAMAP2 adds the identification of complex activities. UCI-HAR contains accelerometer and gyroscope data, PAMAP2 further incorporates magnetometer data on top of that, and WISDM uses only accelerometer data. In addition, the three datasets differ in sampling rate, ranging from 20 Hz to 50 Hz. These differences may impact the performance of the model. Specifically, a higher sampling frequency can capture more dynamic details, thereby helping to improve the model's recognition accuracy; a larger sample size of participants allows the model to learn a wider range of individual differences, thus enhancing its generalization capabilities; and the need for recognizing complex activities requires stronger feature extraction and classification capabilities, which places greater demands on the design and training of the model.

To facilitate fair comparisons with other studies, we employ the same dataset parameters as those used in previous works. To improve the reliability of the experimental results, we adopted the method of repeated experiments and taking the average value. Specifically, each result comes from the average of ten independent runs, each run using a different random initialization. This method ensures the robustness of the results and avoids the problem of local optimal solutions caused by the randomness of a single experiment. The following text provides a detailed introduction to these three datasets.

UCI-HAR [28] The UCI-HAR dataset serves as a collection of smartphone sensor data specifically designed for human activity recognition. It encompasses measurements from the linear accelerometer and gyroscope sensors along the x, y, and z axes, all sampled at a frequency of 50 Hz. The dataset has undergone noise filtering and involves 30 participants aged between 19 and 48 years who performed six standard activities (such as standing, sitting, and stair climbing). The data is segmented into fixed windows of 2.56 seconds (equivalent to 128 data points) with a 50% overlap. For the final composition of the dataset, data from 21 participants were selected for the training set, while the remaining 9 participants constituted the test set, following the default allocation scheme of the dataset.

PAMAP2 [29] The PAMAP2 dataset comprises data collected from nine participants who were equipped with 9-axis *inertial measurement units* (IMUs) placed on their chest, wrist, and ankle. The IMUs sampled data at a frequency of 33 Hz. Participants were instructed to perform a standardized set of activities, including 12 fundamental actions (such as lying down, standing, and stair climbing) and 6 self-selected activities (such as watching TV, driving a car, and engaging in ball sports). To eliminate noise during static periods, data from the 10 seconds before and after each activity were removed. The data was segmented into fixed windows of 5.12 seconds, with a 78% overlap between adjacent windows. For dataset composition, 80% of the data served as the training set, while the remaining 20% constituted the test set.

WISDM [30] The WISDM dataset is constructed from experimental data collected from 29 participants who carried smartphones equipped with three-axis accelerometers in their trouser pockets during daily activities. The data was sampled at a frequency of 20 Hz. Participants were required to perform six designated activities each day, including walking, slow walking, stair ascent, stair descent, standing still, and standing. To ensure data integrity and consistency, any missing values in the dataset were imputed using the corresponding column averages. Data windows were set to a length of 10 seconds, with a 95% overlap between adjacent windows.

For data segmentation, 70% of the data was randomly selected as the training set, while the remaining 30% served as the test set.

3.2. Evaluation metrics

Accuracy and F1-score are both metrics used to evaluate the performance of classification models. Accuracy refers to the proportion of correctly predicted samples by the classification model out of the total number of samples, which is calculated as the number of correctly classified samples divided by the total number of samples. The formula for Accuracy is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where *TP* (True Positive) denotes an instance where the model correctly predicts the positive class, *TN* (True Negative) indicates a correct prediction of the negative class, *FP* (False Positive) represents an incorrect prediction of the positive class, and *FN* (False Negative) signifies an incorrect prediction of the negative class.

The F1-score emerges as a critical metric, encapsulating the harmonic mean of Precision and Recall. Recall is defined as the ratio of accurately predicted positive instances to the total actual positives, whereas Precision is characterized by the ratio of true positive predictions to all positive predictions made by the model. This dual consideration ensures a balanced assessment, capturing the essence of the model's precision in identifying true positives and its sensitivity towards the actual positive cases. The formula for F1-score is as follows:

$$\text{F1-score}=\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{7}$$

Within the realm of classification models, Accuracy finds its utility in scenarios where sample distributions exhibit balance. Conversely, the F1-score comes into play when handling imbalanced datasets. In real-world applications, a holistic evaluation of model efficacy often entails simultaneous scrutiny of both metrics.

3.3. Experimental software and hardware platform

In this study, we employed the PyTorch deep learning framework based on the Windows operating system for training and testing our SHAR models. Our hardware setup utilized a laboratory computer with the following specifications: an NVIDIA RTX 3060Ti *graphics processing unit* (GPU), an AMD Ryzen 5 5600 *central processing unit* (CPU), and 32GB of *random access memory* (RAM).

Building upon successful experiments by previous researchers, we set the number of training epochs for all experiments to 200. As the optimization algorithm, we chose the Adam optimizer with an initial learning rate of 0.001. To adapt to performance variations during training, we dynamically reduced the learning rate by 10% every 50 training epochs. This strategy aims to fine-tune the model's weight adjustments gradually, optimizing the convergence process and enhancing the model's generalization capability. For the loss function, we selected the cross-entropy loss due to its stability and effective gradient information when handling multi-class classification tasks, contributing to more accurate updates during training.

4. Experimental results and analysis

In this research, we embed the prevalent attention mechanisms and our proposed MTSA into the same baseline model across three datasets (UCI-HAR, PAMAP2, and WISDM) to preliminarily evaluate the performance of MTSA. The structure of the baseline model is depicted in Fig. 1. We differentiate MTSA by the number of branches, subdividing it into MTSA_D2, MTSA_D4, MTSA_D6, and MTSA_D8. For instance, MTSA_D2 segments the input features into time slices at two different scales and computes the temporal segment attention within two branch networks. Similarly, MTSA_D4 divides the input features at four distinct scales. Each branch processes the input time series data at a different scale, with a higher number of branches corresponding to finer temporal segmentation. This approach enhances the model's capability to accurately recognize fine-grained activities, as finer temporal resolution can capture more detailed dynamics within the data. However, this improvement in recognition accuracy is accompanied by an increased computational demand, as finer segmentation requires more intensive processing. Therefore, the trade-off between recognition accuracy and computational efficiency is a critical consideration in our experimental design. As shown in Table 1, the performance of MTSA varies across datasets with different branch numbers. In the UCI-HAR dataset, MTSA_D6 significantly outperforms other branch numbers. However, in the other two datasets, MTSA_D8 shows the best performance, not MTSA_D6. This discrepancy is due to MTSA’s sensitivity to the sliding window size of the datasets; a larger sliding window necessitates an increase in MTSA branches to enhance the granularity of feature segmentation. In the PAMAP2 and WISDM datasets, too few branches may lead to performance degradation, as evidenced by MTSA_D2 and MTSA_D4, which underperform compared to the baseline model without any attention mechanism.

Table 1. Comparison of accuracy (%) and F1-score (%) of baseline model with different attention mechanisms on three datasets.

Dataset **Model**	UCI-HAR	PAMAP2	WISDM
Baseline	95.72/95.45	93.30/93.31	98.31/98.24
Baseline+SE	95.76/95.71	92.88/92.86	98.32/98.33
Baseline+CBAM	96.57/96.55	93.47/93.41	98.51/95.49
Baseline+SA	95.28/95.24	**93.66/93.66**	98.24/98.23
Baseline+MTSA_D2	96.17/96.02	92.95/92.96	97.98/97.96
Baseline+MTSA_D4	96.50/96.50	93.24/93.23	98.29/98.29
Baseline+MTSA_D6	**97.18/97.18**	93.34/93.31	98.33/98.32
Baseline+MTSA_D8	96.10/96.07	93.51/93.50	**98.52/98.47**

The baseline model equipped with MTSA mechanism achieves the highest Accuracy and F1-score performance on both the UCI-HAR and WISDM datasets. In the PAMAP2 dataset, MTSA is outperformed by the SA mechanism due to the rich data volume collected using three nine-axis IMUs. SA excels when ample data is available, but its performance declines on the less data-rich UCI-HAR and WISDM datasets. Concurrently, as shown in Table 2, we present the *Floating Point Operations* (FLOPs) and the number of parameters (Param) for each module under consistent input feature size. SA’s computational power consumption is considerably high, whereas MTSA’s computational load and parameter count, although increasing with the number of branches, are negligible compared to other attention mechanisms.

Table 2. Comparison of FLOPs and Param with Different Attention Mechanisms.

Model	**FLOPs**	**Param**
SE	10.368K	2.048K
CBAM	11.072K	0.526K
SA	704.384K	5.2K
MTSA_D2	**0.030K**	**0.014K**
MTSA_D4	0.084K	0.028K
MTSA_D6	0.162K	0.042K
MTSA_D8	0.264K	0.056K

We also explore the integration of MTSA into existing neural network architectures to validate its generalization capabilities and to ascertain whether the optimal number of MTSA branches varies across different network structures. Prior to this, we investigated the impact of embedding MTSA at various points within the network architecture on model accuracy. As shown in Fig. 3, we embed MTSA_D6 into different positions of the residual blocks in ResNet18, including after the convolutional layer, after the BN layer, after the ReLU layer, in the identity mapping branch, and after the addition operation. Figure. 4 demonstrates that, upon evaluating the performance of the MTSA_D6 module embedded at these locations using the UCI-HAR dataset, the model exhibits optimal performance when MTSA is embedded after the ReLU layer. The non-linear properties of the ReLU layer facilitate more efficient gradient propagation to deeper layers of the network, thereby aiding in the effective training and fine-tuning of MTSA parameters.

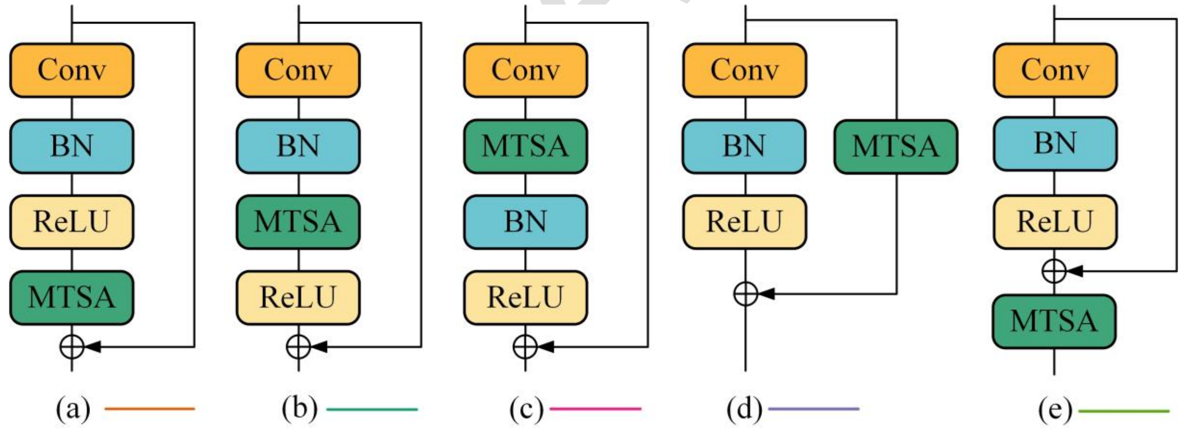


Fig. 3. Embedding MTSA_D6 in different locations of the residual blocks. (a) after the convolutional layer, (b) after BN layer, (c) after ReLU layer, (d) in the identity mapping branch, (e) after the addition operation.

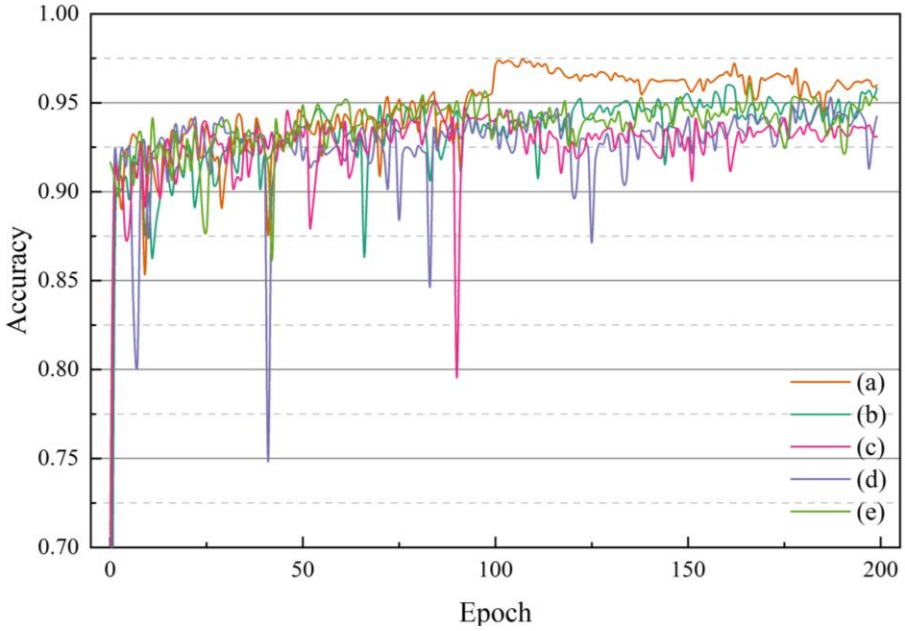


Fig. 4. Accuracy variation curve of MTSA_D6 embedded in different positions of residual blocks trained on UCI-HAI dataset.

Figure 5 provides a detailed analysis of the impact of incorporating MTSA into three different neural network architectures: VGG, ResNet18, and BiLSTM, across two datasets: UCI-HAR and PAMAP2. Given that the UCI-HAR and PAMAP2 datasets cover activity types of varying complexity as well as a wide range of sensor types, we chose these two datasets in this part of the experiment to evaluate model performance. The evaluation metrics considered are model accuracy and F1-score. To accommodate time-series data processing, both the VGG and ResNet18 models underwent modifications. Specifically, the standard 2D convolutional layers were replaced with 1D convolutional layers. Additionally, the BiLSTM model was configured with 256 neurons. In the case of VGG and ResNet18, the MTSA was placed after the activation function to enhance feature representation. For the BiLSTM model, the MTSA was positioned between the BiLSTM layer and the fully connected layer. This strategic placement leverages the contextual information generated by the BiLSTM layer, allowing for more effective identification and utilization of key features, ultimately improving overall model performance and accuracy. Experimental results demonstrate that configuring MTSA with an appropriate number of branches significantly enhances model performance across all three architectures. Notably, combining MTSA with ResNet18 achieved the highest accuracy levels on both datasets. Specifically, in the UCI-HAR dataset with a sliding window size of 2.56 seconds, MTSA_D6 exhibited the most significant performance improvement across different models. Meanwhile, in the PAMAP2 dataset with a sliding window size of 5.12 seconds, MTSA_D8 demonstrated the largest performance gain. These findings align with previous research conclusions, emphasizing the close relationship between the optimal number of MTSA branches and the sliding window size of the dataset.

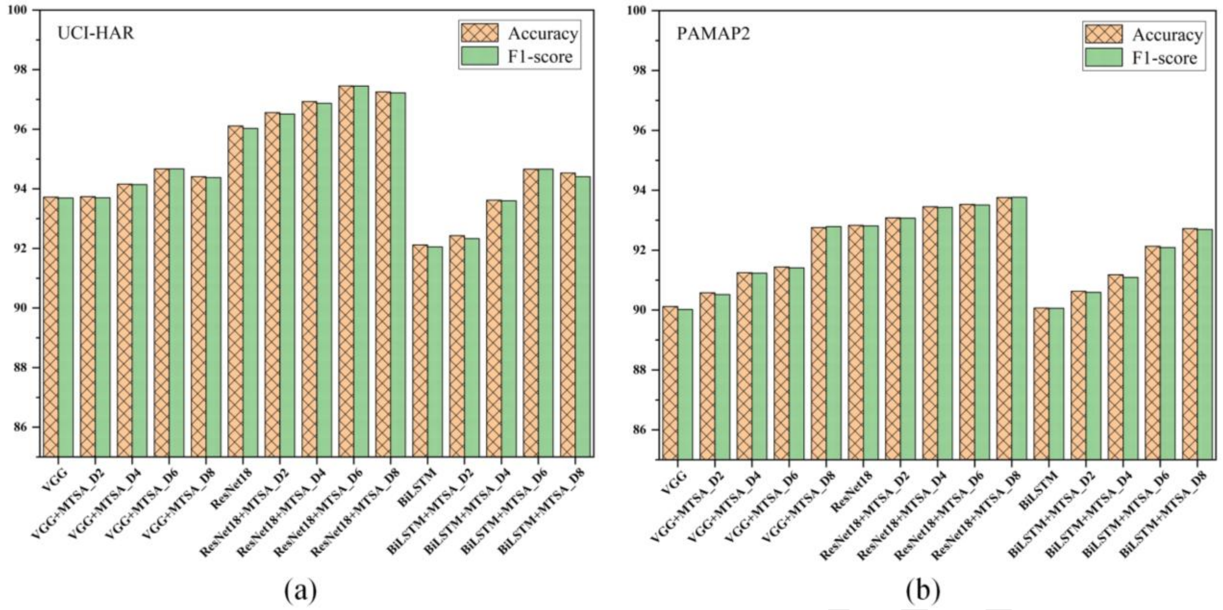


Fig. 5. Comparison of accuracy (%) and F1-score (%) of embedding MTSA into different models.

In this study, we further embed other attention mechanisms into the same location within the same model for comparison. The comparative results are presented in Table 3. We utilize the optimal number of MTSA branches as indicated by previous experimental outcomes, specifically employing MTSA_D6 for the UCI-HAR dataset and MTSA_D8 for the PAMAP2 dataset. According to our experimental results, the analysis under different datasets (UCI-HAR and PAMAP2) and different models (VGG, ResNet18, BiLSTM) shows that the MTSA module significantly improves the performance of the model without increasing the computational complexity and the number of model parameters of the the performance of the model. On the UCI-HAR dataset, whether based on VGG, ResNet18 or BiLSTM architectures, MTSA brings significant performance improvement. In particular, under the ResNet18 architecture, MTSA enabled the accuracy and F1-score to reach 97.46% and 97.45%, respectively, outperforming other enhancement techniques such as SE, CBAM, and SA. This indicates that MTSA has a significant advantage in processing simpler activity recognition datasets such as UCI-HAR.

On the PAMAP2 dataset, MTSA also shows good performance. Although the accuracy improvement of MTSA is not as significant as SA in some cases, such as the ResNet18 architecture, the additional computational complexity brought by SA cannot be ignored. In contrast, MTSA achieves performance gains with little additional computational complexity, a feature that makes it more attractive in resource-constrained application environments.

In Fig. 6, we visually demonstrate the performance changes of the ResNet18 model on the UCI-HAR and PAMAP2 datasets before and after incorporating MTSA. From Fig. 6(a) and Fig. 6(b), it is evident that ResNet18 exhibits varying degrees of misclassification when distinguishing between activities such as ascending and descending stairs or sitting and standing. However, with the inclusion of MTSA, the model accurately identifies instances of misclassification between these activities, further mitigating the issue. In Fig. 6(c) and Fig. 6(d), where a more diverse set of activities is considered, the improvement in recognition accuracy is nearly universal after integrating MTSA. These findings align with prior research conclusions, emphasizing the close relationship between the optimal number of MTSA branches and the dataset's sliding window size.

Table 3. Comparison of Flops, Param, Accuracy, and F1 score between MTSA and other attention mechanisms.

	UCI-HAR			PAMAP2		
Metrics **Structure**	FLOPs (M)	Param. (M)	Accuracy/ F1-score(%)	FLOPs (M)	Param. (M)	Accuracy/ F1-score(%)
VGG	63.68	35.09	93.72/93.69	78.19	39.31	90.12/90.02
VGG+SE	63.72	35.09	94.45/94.39	78.25	39.32	90.03/90.02
VGG+CBAM	63.89	35.17	94.64/94.53	78.42	39.41	92.96/92.91
VGG+SA	76.48	35.94	89.28/89.29	94.83	40.16	92.88/92.79
VGG+MTSA	63.68	35.09	94.67/94.67	78.19	39.31	92.76/92.79
ResNet18	493.48	3.85	96.11/96.03	652.43	3.86	92.83/92.81
ResNet18+SE	493.74	3.86	96.29/96.31	652.74	3.87	93.46/93.45
ResNet18+CBAM	493.92	3.94	97.11/97.09	652.93	3.96	93.64/91.46
ResNet18+SA	605.28	4.72	94.85/94.84	800.24	4.73	**93.87/93.84**
ResNet18+MTSA	493.48	3.85	**97.46/97.45**	652.44	3.87	93.76/93.77
BiLSTM	221.91	1.72	92.12/92.05	297.66	1.75	90.07/90.06
BiLSTM +SE	221.94	1.73	93.64/93.66	297.69	1.76	90.81/90.74
BiLSTM +CBAM	221.96	1.73	94.41/94.41	297.71	1.76	92.14/92.15
BiLSTM +SA	227.35	1.75	93.76/93.74	303.15	1.78	92.97/92.79
BiLSTM +MTSA	221.91	1.72	94.66/94.66	297.66	1.75	92.72/92.69

To further evaluate the robustness and practicality of MTSA, and to ensure consistency in experimental conditions, we replaced the attention mechanisms in the most advanced time-series analysis frameworks with MTSA. We then compared the accuracy and parameter count of the models before and after the replacement. Based on the experimental results, MTSA with varying branch numbers was utilized across different datasets: MTSA_D6 for UCI-HAR, and MTSA_D8 for PAMAP2 and WISDM. The models compared included the following four:

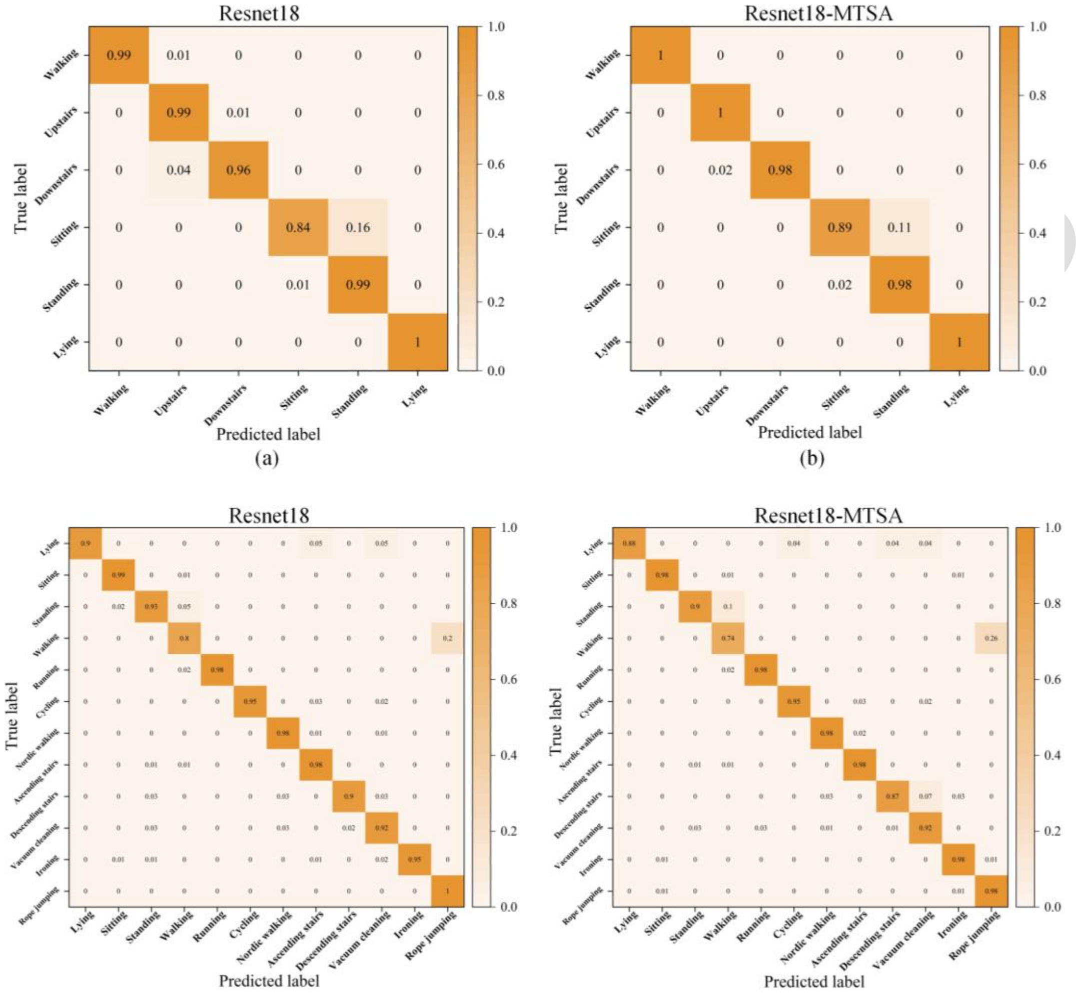
Model 1 [26]: DMEFAM (*Deep Multi-Feature Extraction Framework based on Attention Mechanism*) encompasses a *Temporal Attention Feature Extraction Layer* (TAFEL), a *Channel and Spatial Attention Feature Extraction Layer* (CSAFEL), and an output layer. TAFEL consists of *Bidirectional Gated Recurrent Units* (Bi-GRU) and a *Self-Attention* (SA) mechanism, while CSAFEL is composed of a *Convolutional Block Attention Module* (CBAM) and *Residual Network 18* (ResNet-18). In this study, we substitute the SA mechanism in TAFEL with our *Multi-scale Temporal Self-attention* (MTSA) mechanism and conduct a performance comparison.

Model 2 [12]: CNN-LSTM architecture begins with four sequentially connected convolutional blocks, followed by two layers of LSTM units. Between the LSTM layers, we employ a self-attention mechanism to highlight temporally relevant features. During the comparison phase, we replace the original SA mechanism with MTSA.

Model 3 [31]: TS-DyConv (*Temporal-Spatial Dynamic Convolution*) utilizes an SE module to perform attention calculations on both temporal and spatial dimensions of the convolutional kernels, weighting them to synthesize a novel convolutional kernel. This approach not only diversifies the convolutional kernels but also enhances the model's representational capacity.

In our research, we use standard convolutional kernels combined with the MTSA mechanism to replace the original attention-based kernels.

Model 4 [32]: IDeepConvLSTM, similar to CNN-LSTM, merges Convolutional Neural Networks and LSTM to extract temporal and spatial features. The distinction lies in the embedding of an SE module within the convolutional neural network to emphasize key features. In our study, the MTSA mechanism is utilized to replace the SE module.



(a)

(b)

(c)

(d)

Fig. 6. Comparison of Confusion Matrices Before and After Embedding MTSA in Resnet18.

As Table 4 illustrates, the substitution of the attention mechanism with MTSA resulted in varying degrees of improvement across all models, except for the CNN-LSTM model on the UCI-HAR dataset, which did not show an increase in accuracy. Notably, MTSA does not impose an additional parameter burden on the models; in fact, it often results in a reduction of parameters after replacing the original attention mechanism. It is observed that TS-DyConv is also a remarkably lightweight network. Thanks to the attention computation applied to the convolutional kernels, there is no significant increase in the number of parameters. However, it is known from the original text that the computational load has increased significantly, contributing to longer inference times [31]. After replacing the SA mechanism with MTSA, the CNN-LSTM achieved the best accuracy on the PAMAP2 and WISDM datasets, yet its parameter count is several times larger than that of the other models. This is attributed to its use of oversized convolutional kernels, which provide a larger receptive field and thereby enhance model performance.

Table 4. Comparison of other advanced models before and after replacing MTSA.

	UCI-HAR	**PAMAP2**	**WISDM**
DMEFAM	96.0%/1.60M	92.77%/1.72M	97.9%/1.55M
CNN-LSTM	**98.29%**/42.3M	93.89%/43.1M	98.77%/41.9M
TS-DyConv	97.21%/0.11M	93.04%/0.12M	98.16%/0.11M
IDeepConvLSTM	96.51%/0.98M	92.75%/0.99M	97.14%/0.98M
DMEFAM-MTSA	97.47%/1.11M	93.77%/1.23M	97.94%/1.06M
CNN-LSTM-MTSA	98.14%/36.2M	**94.41%**/37.1M	**98.86%**/35.6M
TS-DyConv-MTSA	97.96%/0.11M	93.79%/0.12M	98.48%/0.11M
IDeepConvLSTM-MTSA	97.55%/0.98M	93.64%/0.98M	98.07%/0.98M

5. Conclusion

To address the issue of multi-class activity mixing when using fixed-size sliding windows to segment datasets in the field of SHAR, and the current limitations of temporal attention mechanisms that tend to borrow spatial attention mechanisms from the field of image classification or natural language processing without optimization for SHAR data characteristics, this study introduces MTSA mechanism. The MTSA considers activities as segments within the data, thus dividing the data into equal-sized temporal segments and calculating statistical information for these segments to determine their attention weights. Moreover, to accommodate different types of activities, the MTSA also integrates attention information from temporal segments of varying sizes. Extensive evaluations on the UCI-HAR, PAMAP2, and WISDM datasets demonstrate that the incorporation of MTSA significantly enhances performance in both baseline and existing model frameworks without adding extra computational burden. This is because MTSA calculates attention weights for temporal segments rather than for each individual timestamp, thereby substantially reducing the computational load. Furthermore, by adjusting the number of MTSA branches, the model’s robustness can be effectively enhanced when dealing with activities of varying complexities or data sampled at different rates. Consequently, MTSA is capable of handling activity recognition tasks across a wide range of scenarios. Notably, the combination of CNN-LSTM with MTSA achieves accuracy rates of 98.14%, 94.41%, and 98.86% on the aforementioned datasets, respectively, positioning them at the forefront of current advancements.

The proposed MTSA module has demonstrated significant performance improvements in multiple experiments, particularly in resource-constrained environments, highlighting its broad potential for practical applications. In the field of health monitoring, MTSA can be applied to sensor data processing in wearable devices to achieve more accurate body activity recognition, while its low computational complexity helps reduce power consumption and extend device battery life. We believe that with further research and development, MTSA will play a key role in future smart devices and services, providing users with smarter, more convenient, and safer experiences. This technological advancement offers new approaches to addressing real-world activity recognition challenges.

Currently, we apply a uniform time interval for the segmentation of all time series. However, the ideal scenario would involve segmenting data from different sensors or axes according to their individual characteristics. In future research, we aim to segment these time series at varying scales to enhance the model's generalization capability.

References

[1] Xu, T., Se, H., & Liu, J. (2020). A two-step fall detection algorithm combining threshold-based method and convolutional neural network. *Metrology and Measurement Systems*, *28*(1), 23–40. https://doi.org/10.24425/mms.2021.135999

[2] Ding, H., Shangguan, L., Yang, Z., Han, J., Zhou, Z., Yang, P., Xi, W., & Zhao, J. (2015). FEMO: A Platform for Free-weight Exercise Monitoring with RFIDs. *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 141–154. https://doi.org/10.1145/2809695.2809708

[3] Deep, S., & Zheng, X. (2019). Leveraging CNN and Transfer Learning for Vision-based Human Activity Recognition. *2019 29th International Telecommunication Networks and Applications Conference (ITNAC)*, 1–4. https://doi.org/10.1109/ITNAC46935.2019.9078016

[4] Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., & Liu, J. (2022). Human Action Recognition From Various Data Modalities: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–20. https://doi.org/10.1109/TPAMI.2022.3183112

[5] Jiao, W., & Zhang, C. (2023). An Efficient Human Activity Recognition System Using WiFi Channel State Information. *IEEE Systems Journal*, *17*(4), 6687–6690. https://doi.org/10.1109/JSYST.2023.3293482

[6] Yao, Q.-Y., Chen, P.-L., & Chen, T.-S. (2023). Human Activity Recognition Using 2-D LiDAR and Deep Learning Technology. *IEEE Sensors Letters*, *7*(10), 1–4. https://doi.org/10.1109/LSENS.2023.3316882

[7] Du, H., Wei, H., Ni, P., Feng, Z., Sun, S., Jiang, M., & Xu, G. (2023). Millimeter Wave Radar Human Activity Recognition with a Contrastive Learning Network. *2023 International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, 1–3. https://doi.org/10.1109/ICMMT58241.2023.10277131

[8] Liu, R., Ramli, A. A., Zhang, H., Henricson, E., & Liu, X. (2022). An Overview of Human Activity Recognition Using Wearable Sensors: Healthcare and Artificial Intelligence. In B. Tekinerdogan, Y. Wang, & L.-J. Zhang (Eds.), *Internet of Things – ICIOT 2021* (pp. 1–14). Springer International Publishing. https://doi.org/10.1007/978-3-030-96068-1_1

[9] Muangprathub, J., Sriwichian, A., Wanichsombat, A., Kajornkasirat, S., Nillaor, P., & Boonjing, V. (2021). A Novel Elderly Tracking System Using Machine Learning to Classify Signals from Mobile and Wearable Sensors. *International Journal of Environmental Research and Public Health*, *18*(23), 12652. https://doi.org/10.3390/ijerph182312652

[10] Bhuiyan, Mohd. S. H., Patwary, N. S., Saha, P. K., & Hossain, Md. T. (2020). Sensor-Based Human Activity Recognition: A Comparative Study of Machine Learning Techniques. *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, 286–290. https://doi.org/10.1109/ICAICT51780.2020.9333470

[11] Khan, R., Abbas, M., Anjum, R., Waheed, F., Ahmed, S., & Bangash, F. (2020). Evaluating Machine Learning Techniques on Human Activity Recognition Using Accelerometer Data. *2020 International Conference on UK-China Emerging Technologies (UCET)*, 1–6. https://doi.org/10.1109/UCET51115.2020.9205376

[12] Meena, T., & Sarawadekar, K. (2023). Seq2Dense U-Net: Analyzing Sequential Inertial Sensor Data for Human Activity Recognition Using Dense Segmentation Model. *IEEE Sensors Journal*, *23*(18), 21544–21552. https://doi.org/10.1109/JSEN.2023.3301187

[13] Teng, Q., Tang, Y., & Hu, G. (2023). RepHAR: Decoupling Networks With Accuracy-Speed Tradeoff for Sensor-Based Human Activity Recognition. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–11. https://doi.org/10.1109/TIM.2023.3240198

[14] Yang, P., Yang, C., Lanfranchi, V., & Ciravegna, F. (2022). Activity Graph Based Convolutional Neural Network for Human Activity Recognition Using Acceleration and Gyroscope Data. *IEEE Transactions on Industrial Informatics*, *18*(10), 6619–6630. https://doi.org/10.1109/TII.2022.3142315

[15] Gao, W., Zhang, L., Huang, W., Min, F., He, J., & Song, A. (2021). Deep Neural Networks for Sensor-Based Human Activity Recognition Using Selective Kernel Convolution. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–13. https://doi.org/10.1109/TIM.2021.3102735

[16] Ishimaru, S., Hoshika, K., Kunze, K., Kise, K., & Dengel, A. (2017). *Towards reading trackers in the wild: Detecting reading activities by EOG glasses and deep neural networks*. 704–711. https://doi.org/10.1145/3123024.3129271

[17] Yao, S., Hu, S., Zhao, Y., Zhang, A., & Abdelzaher, T. (2017). *DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing* (arXiv:1611.01942). arXiv. https://doi.org/10.48550/arXiv.1611.01942

[18] Zhang, J., Liu, Y., & Yuan, H. (2023). Attention-based Residual BiLSTM Networks for Human Activity Recognition. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2023.3310269

[19] Ding, W., Abdel-Basset, M., & Mohamed, R. (2023). HAR-DeepConvLG: Hybrid deep learning-based model for human activity recognition in IoT applications. *Information Sciences*, *646*, 119394. https://doi.org/10.1016/j.ins.2023.119394

[20] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

[21] Duan, F., Zhu, T., Wang, J., Chen, L., Ning, H., & Wan, Y. (2023). A Multitask Deep Learning Approach for Sensor-Based Human Activity Recognition and Segmentation. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–12. https://doi.org/10.1109/TIM.2023.3273673

[22] Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (pp. 3–19). Springer International Publishing. https://doi.org/10.1007/978-3-030-01234-2_1

[23] Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention is All you Need*. Neural Information Processing Systems.

[24] Lu, L., & Deng, T. (2023). A Method of Self-Supervised Denoising and Classification for Sensor-Based Human Activity Recognition. *IEEE Sensors Journal*, *23*(22), 27997–28011. https://doi.org/10.1109/JSEN.2023.3323314

[25] Mekruksavanich, S., Jantawong, P., Phaphan, W., & Jitpattanakul, A. (2024). Hybrid Attention with CNN-BiLSTM and CBAM for Efficient Wearable Activity Recognition. *2024 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 572–576. https://doi.org/10.1109/ECTIDAMTNCON60518.2024.10480050

[26] Wang, Y., Xu, H., Liu, Y., Wang, M., Wang, Y., Yang, Y., Zhou, S., Zeng, J., Xu, J., Li, S., & Li, J. (2023). A Novel Deep Multifeature Extraction Framework Based on Attention Mechanism Using Wearable Sensor Data for Human Activity Recognition. *IEEE Sensors Journal*, *23*(7), 7188–7198. https://doi.org/10.1109/JSEN.2023.3242603

[27] Zheng, G. (2021). A Novel Attention-Based Convolution Neural Network for Human Activity Recognition. *IEEE Sensors Journal*, *21*(23), 27015–27025. https://doi.org/10.1109/JSEN.2021.3122258

[28] Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). *A Public Domain Dataset for Human Activity Recognition using Smartphones*. The European Symposium on Artificial Neural Networks.

[29] Reiss, A., & Stricker, D. (2012). Introducing a New Benchmarked Dataset for Activity Monitoring. *2012 16th International Symposium on Wearable Computers*, 108–109. https://doi.org/10.1109/ISWC.2012.13

[30] Kwapisz, J. R., Weiss, G. M., & Moore, S. (2011). Activity recognition using cell phone accelerometers. *SIGKDD Explor.*, *12*, 74-82. https://doi.org/10.1145/1964897.1964918

[31] Li, Y., Wu, J., Fang, A., & Dong, W. (2023). Temporal-Spatial Dynamic Convolutional Neural Network for Human Activity Recognition Using Wearable Sensors. *IEEE Transactions on Instrumentation and Measurement*, *PP*, 1–1. https://doi.org/10.1109/TIM.2023.3279908

[32] Zhang, N., Song, Y., Fang, D., Gao, Z., & Yan, Y. (2024). An Improved Deep Convolutional LSTM for Human Activity Recognition Using Wearable Sensors. *IEEE Sensors Journal*, *24*(2), 1717–1729. https://doi.org/10.1109/JSEN.2023.3335213

Hailong Rong received the B.E. degree in automation and the M.E. degree in pattern recognition and intelligent systems from Northeastern University, Shenyang, China, in 2003 and 2006, respectively, and the Ph.D. degree in control theory and engineering from Southeast University, Nanjing, China, in 2010. He is currently with the School of Mechanical Engineering and Rail Transit, Changzhou University, Changzhou, China. His research interests are attitude tracking and pattern recognition based on magnetic and inertial measurement units.

Hao Wang obtained a Bachelor's degree in Electrical Engineering and Automation from Changzhou University in 2021. He is currently pursuing a Master's degree in Mechanical and Electronic Engineering from the School of Intelligent Manufacturing Industry at Changzhou University. He mainly engages in research related to inertial sensors.

Xiaohui Wu received his Bachelor's degree in Electrical Engineering and Automation from Anhui University of Technology in 2021. He is currently pursuing his master's degree in the School of Mechanical Engineering and Railway Transportation at Changzhou University, China. His research interests are deep learning-based IMU attitude estimation.

Tianlei Jin received his bachelor's degree in Electrical Engineering automation from Linyi University in 2021. He is currently studying for a Master's degree in Mechanical Engineering and Rail Transportation at Changzhou University, China. His research interests are gyroscopic noise reduction

Ling Zou received the Ph.D. degree in control science and control engineering from Zhejiang University, Hangzhou, China, in 2004. She is currently a professor with the School of Microelectronics and Control Engineering, Changzhou University, Changzhou, China. Her research interests are control engineering, biomedical signal processing, and pattern recognition.