Applied Soft Computing Journal 149 (2023) 110954

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A deep local-temporal architecture with attention for lightweight human activity recognition

Ayokunle Olalekan Ige, Mohd Halim Mohd Noor *

School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia

HIGHLIGHTS

- A new deep parallel architecture that exploits local and temporal features is proposed.
- Each feature learning pipeline has two sub-pipelines to learn local and temporal features of the input window.
- Channel attention is used to increase responsiveness to essential features.
- The size of the model is suppressed using a lightweight neural network module.

ARTICLE INFO

Keywords:
Wearable sensors
Local features
Temporal features
Lightweight
Deep learning

ABSTRACT

Human Activity Recognition (HAR) is an essential area of pervasive computing deployed in numerous fields. In order to seamlessly capture human activities, various inertial sensors embedded in wearable devices have been used to generate enormous amounts of signals, which are multidimensional time series of state changes. Therefore, the signals must be divided into windows for feature extraction. Deep learning (DL) methods have recently been used to automatically extract local and temporal features from signals obtained using wearable sensors. Likewise, multiple input deep learning architectures have been proposed to improve the quality of learned features in wearable sensor HAR. However, these architectures are often designed to extract local and temporal features on a single pipeline, which affects feature representation quality. Also, such models are always parameter-heavy due to the number of weights involved in the architecture. Since resources (CPU, battery, and memory) of end devices are limited, it is crucial to propose lightweight deep architectures for easy deployment of activity recognition models on end devices. To contribute, this paper presents a new deep parallel architecture named DLT, based on pipeline concatenation. Each pipeline consists of two sub-pipelines, where the first sub-pipeline learns local features in the current window using 1D-CNN, and the second sub-pipeline learns temporal features using Bi-LSTM and LSTMs before concatenating the feature maps and integrating channel attention. By doing this, the proposed DLT model fully harnessed the capabilities of CNN and RNN equally in capturing more discriminative features from wearable sensor signals while increasing responsiveness to essential features. Also, the size of the model is reduced by adding a lightweight module to the top of the architecture, thereby ensuring the proposed DLT architecture is lightweight. Experiments on two publicly available datasets showed that the proposed architecture achieved an accuracy of 98.52% on PAMAP2 and 97.90% on WISDM datasets, outperforming existing models with few model parameters.

1. Introduction

The aged and dependent population will pose significant social and economic challenges in the next decades. According to the World Health Organization (WHO), there will be 1.4 billion people 60 and older by 2030, which will increase to 2.1 billion by 2050 [1]. In general, elderly people who are vulnerable because of cognitive and physical limitations need assistance with activities of daily living. However, the cost of having medical staff and caregivers continually watch over elderly people with these issues is a challenge [2]. In recent times, such monitoring has become simpler due to the advancements in ubiquitous computing, which attempts to develop applications running in highly

* Corresponding author.
E-mail address: halimnoor@usm.my (M.H. Mohd Noor).

https://doi.org/10.1016/j.asoc.2023.110954
Received 26 April 2023; Received in revised form 12 October 2023; Accepted 13 October 2023
Available online 20 October 2023

dynamic situations that require minimal human supervision. A typical example is the Human Activity Recognition (HAR) system. HAR systems are designed using external and wearable sensing [3]. Sensors are positioned outside of the person doing the activity in external sensing, while sensors are directly linked to the user or carried around by the user in wearable sensing.

Wearable Sensor HAR can be defined as an approach of seamlessly capturing positional changes of humans using non-infringing devices. Such devices include accelerometers, gyroscopes, and magnetometers, which can be embedded in everyday wearables such as smartphones, smartwatches, smart bracelets, clothing, and shoes, among many others [4]. The most common application of HAR is in pervasive healthcare and rehabilitation. Wearable sensors generate enormous amounts of data, and these signals are multidimensional time series of state changes [5]. Therefore, the signals must first be divided into windows and features extracted to recognize activities. Data from wearable sensors is extracted to train machine and deep learning models. However, with machine learning, feature extraction is hand-crafted and often domain-specific, making feature extraction tedious [6]. For this reason, recent HAR researchers have adopted deep learning to automatically extract features from wearable sensors for activity recognition.

Several deep learning models based on Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been proposed by researchers to learn salient features from wearable sensor signals automatically. For instance, RNN-based models can extract temporal connections and learn features over time intervals, whereas CNN-based models capture the local connections in the current window in activity signals [7,8]. Several activity recognition models have been proposed using CNN, RNNs, or a combination of both for feature learning. For example, the works of Rueda et al. [9], Qi et al. [10], and Bai et al. [11], among others, proposed activity recognition models using CNNs to learn local features, Chen et al. [12], Guan and Plötz [13], and Saha et al., [14] proposed models based on RNNs to extract temporal connections, while Donahue et al. [15], Xia et al. [16], Noor et al. [17], and Park et al. [18], among many others, have proposed models based on hybrid models, which combines CNN with variants of RNNs. Even though these approaches can automatically learn features from wearable sensor signals, tuning deep learning models to capture more discriminative features of human activities is vital.

Generally, wearable sensor HAR involves processing multiple streams of time-series data from various sensors, which can be high-dimensional and noisy, making it a complex task that requires the ability to capture subtle patterns and correlations in the data [19]. Even though shallow architectures can capture relationships between input signals and the target activity, they do not learn hierarchical representations of data, which often limits their ability to capture complex relationships and dependencies of human activity features. For this reason, various works have proposed multiple input deep learning architectures, where a separate network branch processes each input before being combined. However, unlike the proposed DLT, these models often capture local and temporal features on the same heads, which invariably affects the quality of features learned from wearable sensor signals. Also, to improve the quality of learned features, some researchers have introduced attention mechanisms in multiple-input feature learning models, as seen in [8,20–22] and [23], among many others. However, these models often come with large parameters due to the multiple pipelines combined for feature learning, which is unsuitable for wearable computing [24] since resources such as CPU, battery, and memory of such devices are limited.

For this reason, there is a need to propose lightweight deep architectures that can capture more discriminative features of human activities for easy deployment on end devices. To address these challenges, this research proposes a new deep learning architecture that simultaneously learns local and temporal features on different sub-pipelines. The novelty of this work is in the architectural design, which consists of two sub-pipelines concatenated over three independent pipelines to learn local and temporal features simultaneously. The first sub-pipeline uses 1D-CNN to learn local features in the current window, while the second sub-pipeline extracts temporal features using Bi-LSTM and LSTM. Both sub-pipelines are then concatenated before channel attention is added after each concatenation to increase the responsiveness of discriminative features and suppress the less important ones. Then, a global concatenation of the three independent pipelines is done. Specifically, the contribution of this work is in four folds:

i. Firstly, we present a deep learning architecture that simultaneously captures local and temporal features using multiple sub-pipelines to independently learn salient human activity features.
ii. Secondly, the local and temporal features are concatenated along the channel axis, and channel attention is used to increase responsiveness to essential features.
iii. Thirdly, the model size is suppressed by a lightweight neural network module to ensure the model has few model parameters for easy deployment on portable devices.
iv. Lastly, extensive experiments and ablation studies on two publicly available benchmark datasets showed that the proposed DLT model outperformed the existing architectures.

The remainder of this paper is organized as follows: Section II presents a discussion on the related works, Section III presents the methodology of the proposed DLT architecture, Section IV presents the evaluation results and discussion, and Section V concludes.

2. Related works

It is impossible to overstate how essential HAR is to our everyday lives. It has emerged as a topic of interest to scholars from various disciplines [19]. This is because its application cuts across various domains, such as mobile computing [25], context-aware computing [26], ambient assisted living [27], surveillance systems [28], and, most recently, serious games [29]. The most recent deployment of HAR has been in fall detection [30], behavioural monitoring [31], psychological monitoring [32], stress detection [33], and gait anomaly detection [34], among others. Human activity data can be collected using vision-based, radio-based, and sensor-based approaches [3,35,36]. However, the limitations of the vision and radio-based methods have led to the adoption of the sensor-based approach, with wearable sensors being the most adopted due to their advantages over other sensor-based approaches [19].

In wearable sensor-based activity recognition, the sensors are attached to subjects so they can still perform all necessary activities without infringements. Examples of wearable sensors include accelerometers, magnetometers, gyroscopes, and others. Recent advancements in miniaturization have seen these sensors embedded into clothing, shoes, wristwatches, eyeglasses, smartphones, smart belts, smart socks, and smart bracelets, among others [37]. According to a study in [38], shipments of wearable devices, such as wristbands, watches, smartwatches, and others, reached 34.2 million units in the second half of 2019, a 28.8% increase over the previous year. Therefore, human activity recognition researchers easily accept the concept of sensor deployment on wearable devices. Human activities can be divided into basic and complex activities (activities of daily living). Basic activities can be further divided into static, dynamic, and transitional activities, including sitting, standing, sit-to-stand, stand-to-sit, walking, and running, among many others. In contrast, complex activities are interleaves of two or more basic activities, which can involve preparing a meal, shopping, riding a bus, or driving a car.

Literature shows that using features instead of raw data improves classification accuracy [39]. In the literature, several activity recognition models have been trained using machine learning methods, as seen in [40] and [41], among many other works. However, before machine learning techniques can be used for activity recognition, features of the

data must be extracted. This feature extraction method in machine learning is hand-crafted and often domain-specific, making feature extraction tedious [6]. For this reason, recent HAR researchers have adopted deep learning to extract features from wearable sensors for activity recognition. Several researchers have used CNN, RNN, or a hybrid of both methods for feature learning. A discussion of some of these approaches is presented in the sub-sections.

2.1. CNN models

CNN is the most widely used deep learning method for automatically extracting features in human activity identification. This is due to the hierarchical structure of activities, translation invariance, temporally linked readings of time-series signals, and issues with HAR feature extraction. By leveraging multiple-layer CNN with alternating convolution and pooling layers, features are extracted automatically from raw time-series sensor data [42]. Generally, the lower layers of the convolution extract more basic features, and higher layers extract more complex features.

The pioneering research that leveraged CNN for automatic feature learning in sensor-based activity recognition is in Zeng et al. [43], where a single channel CNN layer with partial weight sharing was used to learn discriminative features from accelerometer data. Also, for time series data in general, Zheng et al. [44] proposed a multi-channel deep CNN model. In [45], a multiple-layer CNN model was proposed for human activity recognition, and the model was able to achieve improved recognition performance, especially on dynamic activities, compared to the performance of some shallow models.

A CNN model that analyses each wearable sensor data individually was suggested by Rueda et al. [9]. A dataset used in industry was tested together with two publicly accessible datasets. The model's accuracy of recognition increased for a few specific activities. Qi et al. [10] proposed a deep convolutional neural network model for activity recognition. The accelerometer, gyroscope, and magnetometer data were used to create the model, which included several signal processing algorithms and a signal selection module to increase the accuracy and richness of the raw data. The classification accuracy of experiments on the gathered dataset was 95.27%. The model, however, could not extract quality features for some of the 12 activities, as 5 had low precision and recall in the 50–70% range.

Huang et al. [46] presented a two-stage end-to-end convolutional neural network to improve the quality of the features being extracted from activities such as walking upstairs and downstairs. The model improved recognition accuracy on the two activities compared to a single-stage CNN. Even though the model exceeded the performance of the single-stage CNN, which served as the baseline model, the quality of the features extracted from the activities was still low. In order to improve the feature representability of CNN on wearable sensor datasets, Ahmad & Khan [47] proposed a multistage gated average fusion model, which extracts and fuses features from all the layers of CNN to learn quality features from wearable sensor data. However, the quality of the features extracted was still relatively low. A limitation could be attributed to the long-term dependency of the time series data, which CNN cannot handle.

Since wearable sensors come in time series format, extracting the long-term dependency of the time series using CNN makes it challenging to improve the performance of the activity recognition models, as CNN mainly captures the local features in the current window [48]. Since CNN ignores the temporal dependencies of activity features, some researchers have proposed RNN models for automatic feature learning in activity recognition. The RNN can remember early information in the sequence data and is suitable for processing time-series data.

2.2. RNN models

RNN models can capture temporal information from sequential data and retain temporal memory of signals in a time series. Therefore, they can address the issue of sequential human activity recognition [49]. RNNs consist of the input layer, hidden layers with multiple nodes, and the output layer. RNNs typically experience explosive and disappearing gradient issues. Due to this, the network cannot accurately represent long-term temporal relationships between input signals and human activities. By swapping out conventional RNN nodes with LSTM memory cells, RNNs based on LSTM address the limitations of the traditional RNN nodes and can model lengthy activity windows. In LSTMs, there are four interconnected layers in the repeating module. These layers consist of the cell state layer and three additional levels known as gates. The LSTM unit may decide whether new data should be added to the current memory or if it should be kept. Therefore, LSTM-RNN can create long-range dynamic dependencies to avoid the vanishing or exploding gradients problem while training. In time series classification, the principal elements of an LSTM network include the sequence input layer, the LSTM layer, the fully connected (FC) layer, and the classification output layer with SoftMax. In Edel & Köppe [50], a binarized RNN model was presented, termed a Bidirectional Long Short-Term Memory Recurrent Neural Network (BiLSTM-RNN). The model was benchmarked on two publicly available datasets and one custom dataset. The result showed that the model addressed the problems of bulky model size problems at the expense of high model training time.

Agarwal & Alam [49] proposed a model with two LSTM layers for feature learning in human activity recognition, with each layer in the model having 30 neurons. The model was evaluated on the Wireless Sensor Data Mining (WISDM) dataset using a 180 sliding window size and achieved a recognition performance of 95.78%. Even though the model was less bulky, it still misclassified the walking, walking upstairs, and walking downstairs features, all of which have inter-class similarities. The authors in Barut et al. [51] employed a multi-task LSTM model for activity recognition and intensity estimation after initially developing a new dataset with a single wearable sensor attached to the waist. The authors considered sitting, laying down, standing, walking, walking upstairs, downstairs, and running and used a sliding window segmentation size of 100. However, the computation time was high, and the quality features of some activities were not well learned. In recognizing human activities, processing time is a crucial consideration. This is because most of the activity recognition use cases need immediate performance. Hence, using RNN models for activity recognition is unsuitable for real-world deployment. Recently, some researchers have coupled the feature extraction capabilities of RNNs with the capability of CNN to simulate temporal dependencies among human activities to extract more high-quality features of human activities from wearable sensor signals with minimal computation time.

2.3. Hybrid models

In a move to improve feature learning, some researchers have combined CNN with RNNs to learn temporal and local features from wearable sensor signals. For example, C. Xu et al. [52] proposed InnoHAR, a model that employed 2D-CNN and GRU to improve the quality of features learned from wearable sensor signals. The authors used a sliding window size of 170 on the PAMAP2 dataset with 78% overlap and achieved recognition accuracy of 93.5%. However, the model took around 153 s for activity prediction, and the issue of inter-class similarity was not addressed. In [53], a model based on simple recurrent units (SRUs) with the gated recurrent units (GRUs) of neural networks was proposed. The ability of the SRUs' internal memory states was utilized by the authors to process sequences of multimodal input data and used the deep GRUs to store and learn how much of the previous information is delivered to the future state to solve vanishing gradient difficulties and accuracy fluctuations. Experiments were done on the MHealth dataset, which consists of 12 activities.

Dua et al. [48] merged CNN and GRU in their multi-input hybrid model by combining three CNN-GRU architectures. The model was

evaluated on the PAMAP2, UCI-HAR, and WISDM datasets and achieved 95.27%, 96.20%, and 97.21% accuracies on the three datasets, respectively. However, the model size was relatively large, with high training time. Challa et al. [54] used Time distributed CNN with Bidirectional-LSTM (Bi-LSTM) to categorize multi-activities on the PAMAP2, WISDM, and UCI-HAR datasets, and a sliding window size of 128 was used on the three datasets. The time-distributed CNN had 64 and 32-channel dimensions, with filter sizes 3, 7, and 11. The model achieved an accuracy of 94.27% on PAMAP2, 96.04% on WISDM, and 96.31% on UCI-HAR datasets. However, some activities had precision and recall as low as 70%.

Nafea et al. [55] proposed a CNN-Bi-LSTM model that employs bi-directional long short-term memory and CNN with varied kernel sizes to learn features at various resolutions. Features were extracted using the stacked convolutional layers, and a flattened layer was added before a fully connected layer. Also, another feature learning pipeline with a Bi-LSTM layer and LSTM layer were stacked. The features were also flattened, and then a fully connected layer was added. Subsequently, the features in the fully connected layers were concatenated, followed by another flattened layer before activity classification. The model was evaluated on WISDM and UCI-HAR datasets, and the researchers chose a sliding window size of 128 to segment the signals. Results showed that the model achieved improved classification accuracy. However, the model size was bulky due to the architecture employed in stacking the convolutional layers.

A Bi-LSTM and residual block model was proposed for feature learning in [56]. The model functioned by automatically extracting local features from inputs of multidimensional inertial sensors using the residual block, retrieving the forward and backward dependencies of the feature sequence using Bi-LSTM, and then feeding the features into the Softmax layer for classification. The model was evaluated on PAMAP2 and WISDM and achieved a classification performance of 97.15% and 97.32% on each dataset. In [17], a Conv-LSTM model that uses the sliding window relationship and the temporal features of sensor-based activity recognition data was proposed for salient feature learning. The model concatenated window characteristics, employed a sequence-learning module to learn temporal information, and achieved a 91.6% accuracy on the benchmarking dataset.

Lu et al. [57] proposed a multi-channel CNN-GRU feature learning model for activity recognition. Each channel in the model had two 1D-CNN layers with 64 and 128 channel dimensions and a fixed filter size of 3, 5, and 7 in each channel. The features were concatenated before adding two GRU layers with 128 and 64 neurons. The model was also benchmarked on PAMAP2, WISDM, and UCI-HAR datasets using various sliding window sizes and achieved an accuracy of 96.25%, 96.41%, and 96.67% on the datasets, respectively. In [58], an ensemble of activity recognition models was proposed. The authors developed four standalone feature learning pipeline models and ensembled them to increase feature learning. The four ensembled models consist of a CNN model, an LSTM concatenated with a CNN model, a ConvLSTM model, and a Stacked LSTM model. Prediction using the Ensem-HAR model was achieved by stacking predictions from each previously described model, followed by training a Meta-learner on the layered prediction, which yields the final prediction on test data. The model achieved a classification accuracy of 98.70% on WISDM, 97.45% on PAMAP2, and 95.05% on UCI-HAR. However, the model was highly bulky due to the ensemble of four standalone models. Generally, one limitation of CNN is that it treats all features equally, and since some features in wearable sensor data are often more important than others, some researchers have proposed advanced attention models to increase the responsiveness of activity recognition models to essential features.

2.4. Feature learning with attention mechanisms

The primary idea behind the attention mechanism is to provide various weights to various sorts of information. Consequently, the deep learning model is drawn to it when relevant data is given a higher priority weight [59]. In recent times, researchers have adopted attention mechanisms in HAR. For example, Murahari & Plotz [60] proposed a DeepConvLSTM model with attention to exploring relevant temporal features. A relative improvement of 87.5% was recorded on the PAMAP2 dataset using the model with attention and an accuracy of 74.8% on the model without attention. H. Ma et al. [22] proposed a model called Attnsense. The model combined attention with CNN and GRU to improve salient feature learning from signals from multiple streams. The model achieved an 89.3% F1-score on PAMAP2, with an increased model size, and had a high training time. This can be attributed to the method of having the CNN and GRU on the same heads. In H. Zhang et al. [61], the authors exploited the multi-head approach integrated with attention for human activity recognition. The features were learned using multi-head CNN and concatenated to produce a single feature vector. Thirty parallel attention heads were then used to learn crucial features for precise activity recognition during the feature selection phase. Additionally, the model had about 2.77 million parameters with an F1-score of 95.40% on the WISDM dataset. Zhang et al. [62] proposed another multi-head CNN model for feature learning and induced attention mechanism into each head to address the limitations of the high number of parameters while learning discriminative features. The model was evaluated on two public datasets, and the results showed that the model outperformed the baseline CNN, baseline LSTM, and baseline ConvLSTM that were assessed against the model.

In Khan and Ahmad [21], three convolutional heads designed using one-dimensional CNN were proposed, with each head induced with an attention mechanism. The authors leveraged the squeeze and excitation block presented in Hu et al. [59] as an attention mechanism, placing the block after the first convolutional layer before adding another convolutional layer. The model was tested on the publicly available WISDM and UCI HAR datasets, with a sliding window size of 200 on WISDM and 128 on UCI HAR. The result showed that the model was able to learn improved features. However, the size of the activity recognition model was still relatively large at 1.0415 M, even though it was lower than the model presented in [61]. In [63], a lightweight feature learning model was proposed, which used squeeze and excitation block with best-fit reduction ratio. The SE block was placed after the output of the flattened layer was reshaped, and the model adaptively selected the number of neurons and the reduction ratio in the SE block, then benchmarked on PAMAP2, WISDM, and UCI-HAR datasets, achieving 97.76%, 98.90%, and 95.60% respectively. However, since the parameters on the model were chosen adaptively, the model had 0.549 M on the PAMAP2 dataset, while the size of the model on WISDM and UCI-HAR was quite parameter-heavy.

Xiao et al. [64] proposed a perceptive extraction network to extract salient features from wearable sensor signals using CNN and LSTM with attention. The model stacked three convolutional layers with LeakyReLU activation while using 128 channel dimensions and varying kernel sizes of 5, 7, and 11 in the convolutional layers. The features extracted by this layer were then concatenated with another feature learning pipeline of two 64-neuron LSTM layers with attention before a fully connected layer was added to classify the activities. The model was tested on PAMAP2, UCI-HAR, WISDM, and Opportunity and achieved improved recognition accuracy. However, the model's size was still relatively large, and the model was not able to capture quality discriminative features. In [65], a module termed WSense, which is capable of learning salient features using lightweight models, was proposed and evaluated on PAMAP2 and WISDM datasets. The module was presented as a plug-and-play network, which can be plugged into HAR architectures for parameter reduction, regardless of the sliding window segmentation.

In Mim et al. [8], a GRU inception-attention model was proposed, which used GRU along with Attention Mechanism for the temporal feature learning and Inception module along with Convolutional Block Attention Module (CBAM) for the spatial part of their model.

Experiments showed that the model could not learn features of activities with inter-class similarity. Unlike previous works, our research proposes a DLT architecture that learns local features in the current window using one-dimensional CNN and temporal features using Bi-LSTM and LSTMs over multiple concatenated sub-pipelines. The squeeze and excitation block is then leveraged to boost responsiveness to discriminative features after concatenation, while the WSense module is plugged into the top of the DLT feature learning pipeline to ensure the model size is lightweight.

3. Proposed methodology

Previous feature learning models take no advantage of learning local and temporal features simultaneously on different sub-pipelines. Also, multi-head deep feature learning models often come with high model parameters due to the architecture that combines multiple pipelines. By extracting the local features in the current window on a different sub-pipeline and the temporal features on another, the benefit of CNN and RNNs can be harnessed equally in capturing more discriminative features from wearable sensor datasets. The workflow of the proposed model is presented in Fig. 1.

As shown in Fig. 2, signals collected using wearable sensors were segmented into windows using the fixed sliding window with a degree of overlap, which the DLT architecture takes as inputs before features of the activities are learned and classified. The descriptions of the methods are presented in the sub-sections.

3.1. Sliding window segmentation

The activity signals in this research are segmented using a fixed sliding window with a degree of overlap, as shown in Fig. 2 and further explained.

Given a stream of values (samples) $x_i \in R$ at Time $t_i, i = 0, \ldots, N$, where N is the total number of samples. It is assumed that $t_0 = 0$, and that the period of sampling is constant at ΔT, such that;

$$\Delta T = t_{i+1} - t_i \tag{1}$$

Using a fixed sliding window size, the signals are split into segments of n samples where $n > 1$. Therefore, the window size w can be given as:

$$w = n\Delta T \tag{2}$$

Typically, the segmentation is performed with a degree of overlapping. Given $m \in \{1, 2, 3 \ldots, n-1\}$ as the number of samples in a certain overlapping period between two consecutive sliding windows, the overlapping period between two consecutive windows in seconds is such that:

$$v = \quad m\Delta T \tag{3}$$

where the overlapping period is considered as a percentage of the total length of the window and is given as:

$$y(\%) = \frac{m}{n} \tag{4}$$

The overlapping is needed to increase the segmentation numbers to allow better generalization of activity recognition models. Hence, each sliding window $W_k^c, k = 0, \ldots K$ can, therefore, be given as a set of samples x_i, such that:

$$W_k^c = \left\{ x_{k(n-m)} \quad , x_{k(n-m)+1}, x_{k(n-m)+2}, \ldots, x_{k(n-p)+n-1} \right\} \tag{5}$$

where c is the data channels of the sensors, and K is the total number of sliding windows.

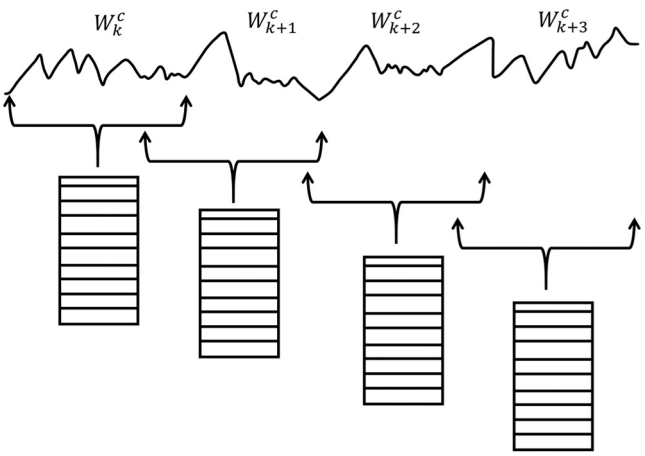

Fig. 2. Fixed sliding window with overlap.

3.2. Deep local-temporal model

The proposed architecture consists of two sub-pipelines

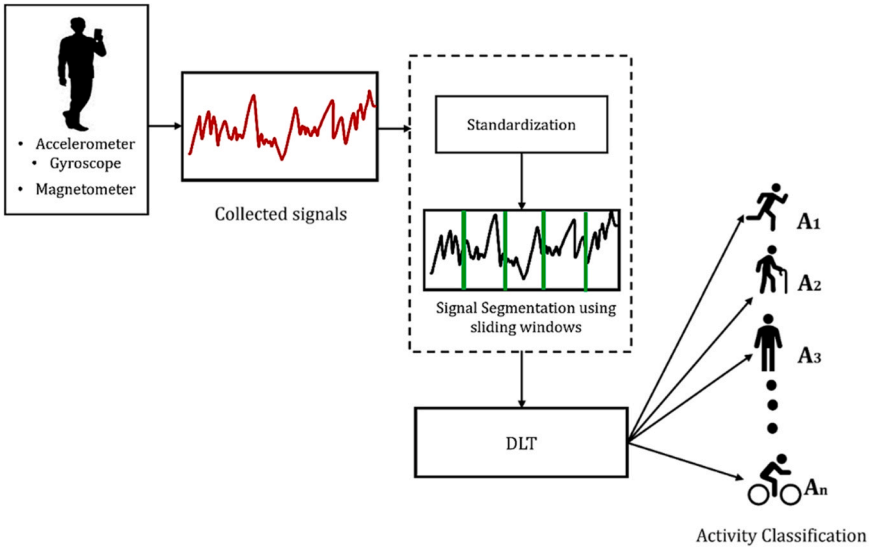

Fig. 1. Workflow of the proposed DLT architecture.

concatenated over three pipelines to learn local and temporal features simultaneously. The first sub-pipeline uses 1D-CNN to learn local features in the current window, while the second sub-pipeline extracts temporal features using Bi-LSTM and LSTM. Both sub-pipelines are then concatenated before the SE block is added after each concatenation to increase responsiveness to discriminative features, and then a global concatenation of the three pipelines is done. The architecture of the proposed DLT model is presented in Fig. 3 and further discussed.

3.2.1. Feature learning pipelines

Each feature leaning pipeline in the DLT consists of two sub-pipelines of 1D-CNN, which captures the local features in the current window, and the Bidirectional LSTM and LSTM layers, which capture the temporal features. In extracting the local features, the segmented data was passed to 1D-CNN layers with 3, 5, and 7 kernel filters and 16, 32, and 64 channel dimensions, with ReLU activation function. By employing progressively increasing kernel sizes (3, 5, and 7), the model becomes adept at detecting patterns of various scales within the input data. The smaller kernels capture the finer details, while the larger kernels grasp broader trends. This multiscale analysis ensures that the model can capture a wide spectrum of features. Also, the ascending channel dimensions (16, 32, and 64) correspondingly increase the complexity and depth of feature extraction. This hierarchical abstraction enables the 1D-CNN layers to learn more intricate and higher-level features progressively. Each layer learns local features of the input data, builds feature maps based on convolutional filters, and recognizes intrinsic features in the output of the layer before it. A Batch Normalization layer is used to speed up learning and prevent covariate shift issues before a maxpool layer is added. LSTM and Bi-directional LSTM layers are leveraged to extract the temporal features. The LSTM layer consists of LSTM units and a shared architecture of these units, namely an input gate, output gate, cell, and forget gate. The architecture of the LSTM is presented in Fig. 4.

As shown in Fig. 4, H_t, F_t, C_t, and O_t denotes the hidden state, the forget state, the memory cell state, and the output. x_t denotes the input at time step t. The block contains sigmoid and tanh functions. By using a forget mechanism, the LSTM network's initial operation seeks to specify the data to be captured from the previous hidden state, which can be expressed as presented in Eq. (6).

$$F_t = \sigma\left(W_f x_t + b_f + U_f H_{t-1} + c_f\right) \tag{6}$$

where W_f, U_f, b_f and c_f are the weights and biases of forget gate. $F_t = 1$ means all previous hidden state is preserved, and $F_t = 0$ means all previously hidden state information is cleared. The next operation, which uses two mechanisms, decides how much of the new input should be preserved. Eq. (7) describes how the input gating determines what needs to be updated first. Second, the tanh function determines the likely state value, as shown in Eq. (8).

$$I_t = \sigma(W_i x_t + b_i + U_i H_{t-1} + c_i) \tag{7}$$

$$G_t = \tanh\left(W_g x_t + b_g + U_g H_{t-1} + c_g\right) \tag{8}$$

where W_i, W_g, U_i, U_g, b_f, b_g, c_i and c_g are the weights and biases of input gate. The current cell state data is then calculated as stated in Eq. (9).

$$C_t = F_t * C_{t-1} + I_t * G_t \tag{9}$$

where $*$ is the element-wise multiplication. Finally, the hidden state H_t is calculated by applying the tanh function to the computed memory state C_t, with the output gate O_t influencing the information retained in the hidden state, and it is shown in Eq. (10).

$$O_t = \sigma(W_o x_t + b_o + U_o H_{t-1} + c_o) \tag{10}$$

The hidden state H_t is then expressed as Eq. (11), where $H_t \in R^d$, and d is the dimension of the features.

$$H_t = \tanh(C_t) * O_t \tag{11}$$

Since LSTM layers extract features in only one direction, the segmented data in the DLT model was passed to a Bi-directional LSTM layer. The LSTM layers in the forward and backward layers of the BiLSTM collectively determine the output of the BiLSTM layer. The structure of the BiLSTM layer is presented in Fig. 5.

The output layer y_t of the BiLSTM is expressed as shown in Eq. (12).

$$y_t = \left[\overrightarrow{H}_t, \overleftarrow{H}_t\right] \tag{12}$$

where $H, \overleftarrow{H}_t$ is the forward and backward result of the LSTMs and y_t is the concatenated result of the LSTM units. By using the BiLSTM, faster and richer features can be learned. In the DLT model, two BiLSTM layers are stacked to improve the quality of the learned temporal features, with one-dimensional maxpool layer between them, before a single LSTM layer is passed, and another maxpool layer is added.

3.2.2. Sub-pipeline concatenation and feature weighting

After the local features in the current window and the temporal features have been extracted using the two sub-pipelines, a concatenation layer is then used to concatenate the features in the maxpool layer of the local feature learning sub-pipeline, with the maxpool layer of the temporal feature learning sub-pipeline, along the channel dimension. Then, the squeeze and excitation (SE) block, presented in Fig. 6, is placed to recalibrate the features, such that important feature maps are emphasized while suppressing less important ones using channel weights. It is especially effective in improving the information flow within a network by adaptively recalibrating channel-wise features.

The SE block consists of two main steps: squeezing and exciting. In the squeeze step, the global information is gathered from the channel-wise feature maps. Each channel's information is compressed into a single number by applying global average pooling (GAP). This pooling operation averages the values in each channel to obtain a scalar representation. In the excite step after obtaining the global information, the excitation step involves learning a set of channel-specific weights (parameters) representing each channel's importance. This is often done using one or more fully connected layers or convolutional layers with non-linear activations. These weights determine how much each channel's information should be amplified or suppressed. Then, the SE block's output is obtained by multiplying the original feature maps by the learned channel weights. This process effectively scales the feature maps according to the learned importance of each channel.

In the proposed model, the aggregated information about the features in each concatenated sub-pipeline (channel statistics) $\boldsymbol{u} \in \mathbb{R}^{L \times D}$ is obtained by passing the concatenated feature maps to the GAP layer of the SE block. Therefore, generating the statistic $z \in R^D$ by squeezing $\boldsymbol{u}$ through L. Hence, the d-th element of z is given as:

$$z_d = F_{sq}(\boldsymbol{u}_d) = \frac{1}{L}\sum_{j=1}^{L} u_d^j \tag{13}$$

where F_{sq} is given as the squeeze function and L is the length of the feature maps, D is the number of output filters or feature maps generated by the concatenation.

The aggregated information acquired using the squeeze operation is then passed to the excitation operation to capture channel-wise dependencies using a gating mechanism with a sigmoid activation function, given as:

$$\boldsymbol{s} = F_{ex}(\boldsymbol{z}, W) = \sigma(g(\boldsymbol{z}, W)) = \sigma(W_2 \delta(W_1 \boldsymbol{z})) \tag{14}$$

where σ is the sigmoid activation function, and δ is the ReLU activation function, $\boldsymbol{z}$ is the input to the excitation operation, $W_1 \in \mathbb{R}^{D/r \times D}$, $W_2 \in \mathbb{R}^{D \times D/r}$ are the weight vectors, and r is the reduction ratio. s is a vector of

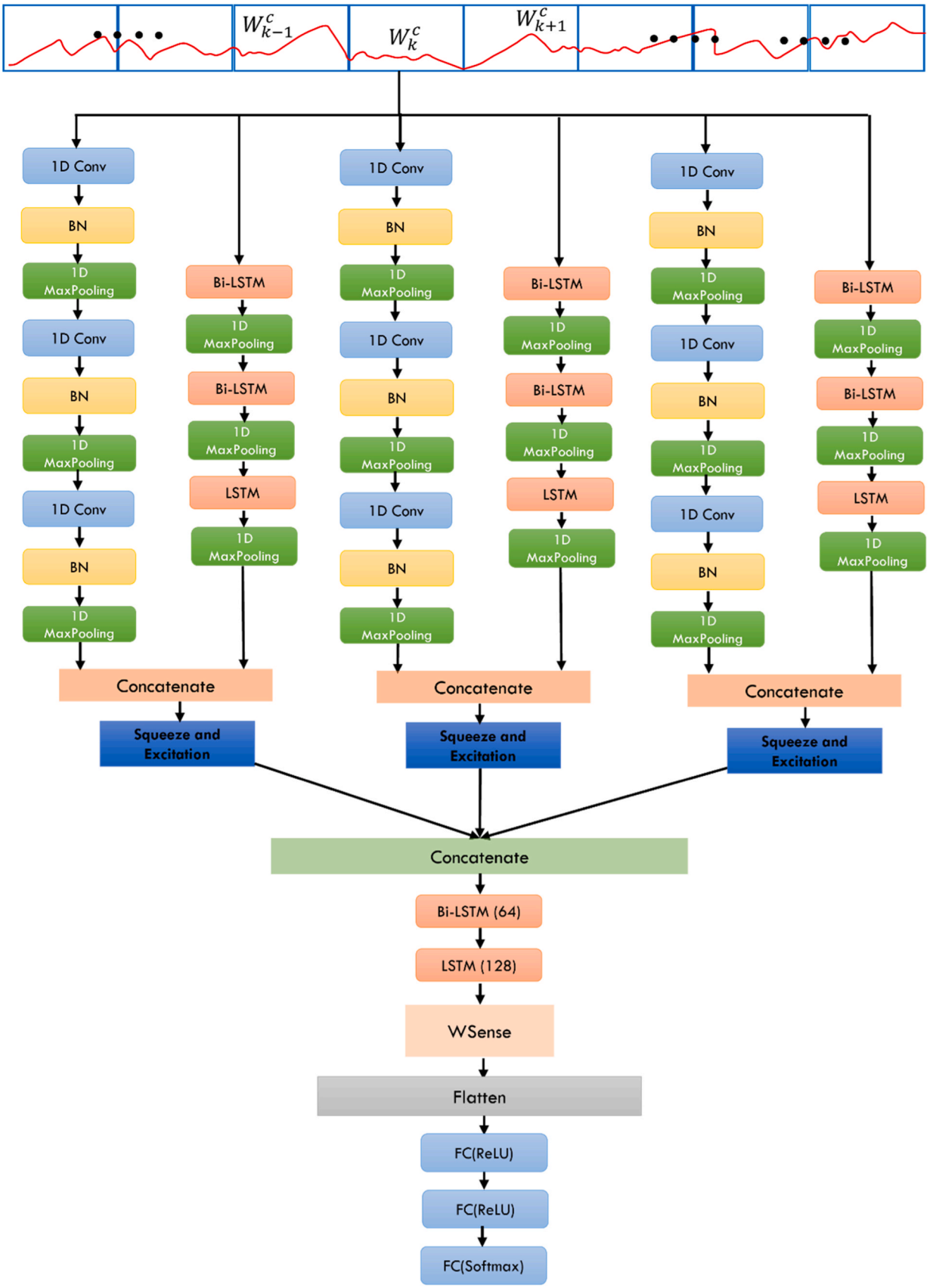

Fig. 3. Architecture of the Deep Local-Temporal Feature Learning model.

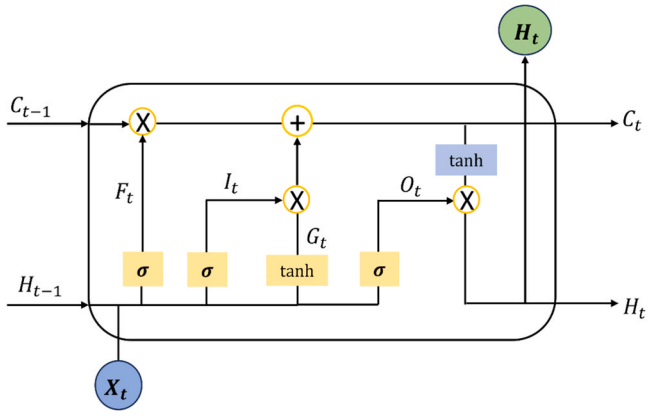

Fig. 4. Architecture of the LSTM.

size equal to the number of feature maps. Thus, the values can be interpreted as the weights indicating the importance of the feature maps. Using s, the feature map u_d is rescaled as follows:

$$F_{(scale)}(u_d, s_d) = s_d . u_d \tag{15}$$

where $F_{(scale)}(u_d, s_d)$ is the channel-wise multiplication of a scalar s_d and feature map u_d.

3.2.3. Pipeline concatenation and model size reduction

In the DLT model, the process is replicated over two additional pipelines, taking the total sub-pipelines to 3 local and 3 temporal feature learning sub-pipelines, and the discriminative features in the SE Block are then concatenated along the channel axis. After concatenation, one Bi-LSTM layer with 64 neurons and an LSTM layer with 128 neurons are added to retain the sequence of the features learned before the WSense module presented in [65] is added to reduce the model size and learn more salient features. In the WSense, a 1D convolutional layer takes in the feature maps in the LSTM layer and then used a global max pooling layer to downsample the input, taking the maximum value over each feature map. A second 1D convolutional layer is then added to detect the local conjunctions in the preceding feature maps, using a 1 kernel size and sigmoid activation function. After this, the maximum value over each feature map in the first convolutional layer of the WSense is then calibrated with the features in the second convolutional layer using an element-wise multiplication. A flatten layer was then added before including two fully connected layers with ReLU activation function. Lastly, a fully connected layer with a softmax activation function is added for activity classification. The probability of activity class i is given as:

$$g(z)_i = \frac{e^{z_i}}{\sum_j^K e^{z_j}} \tag{16}$$

where z values represent the model's computed scores for each class, j is the index that iterates over all possible classes, typically from 1 to K, and K is the total number of classes.

4. Results and discussion

This section presents the datasets used for model evaluation, the flow of experiments, and the results and discussion on the evaluation results.

4.1. Datasets

4.1.1. PAMAP2

The PAMAP2 dataset [66] has nine participants who were required to participate in eighteen (18) activities. These activities included 12 protocol activities performed by all the subjects and six (6) optional activities performed by some subjects. The activities include sitting, standing, running, descending stairs, ascending stairs, cycling, walking, Nordic walking, vacuum cleaning, computer work, car driving, ironing, folding laundry, house cleaning, playing soccer, and rope jumping. Gyroscopes, accelerometers, magnetometers, heart rate monitors, and temperature measurements were used for data collection. This research considered the protocol activities and 36 features of 3 IMUs, including accelerometers, gyroscopes, and magnetometers.

4.1.2. WISDM dataset

The WISDM dataset [67] is an activity recognition dataset gathered from 36 participants who go about their daily lives. Accelerometer data from the three-axis was considered. The dataset consists of 6 activities: walking, sitting, standing, jogging, ascending, and descending stairs. The data was collected at a 20 Hz sampling rate using a smartphone accelerometer sensor.

4.2. Experimental design

Experiments on the DLT architecture were carried out in nine phases, as shown in Fig. 7.

The first set of control experiments concatenated one local and one temporal (1 L-1 T) feature learning pipeline, which was then used to classify activities directly before evaluation. The second experiment

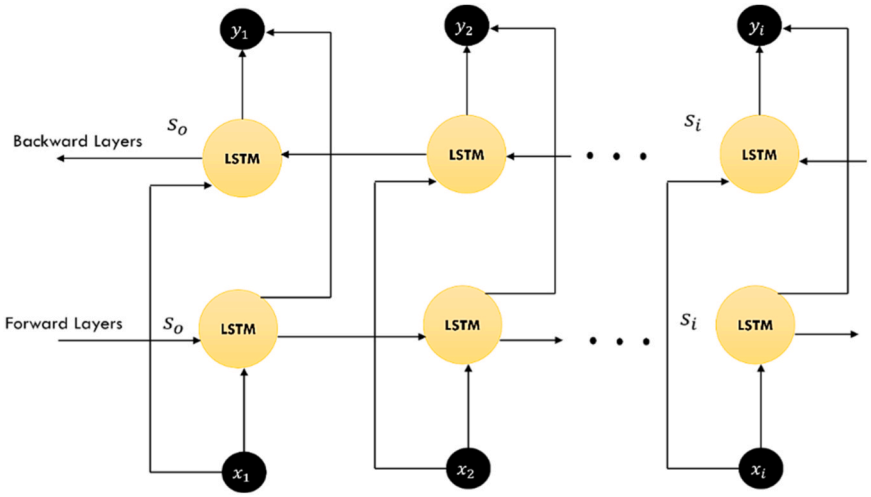

Fig. 5. Architecture of the Bi-LSTM.

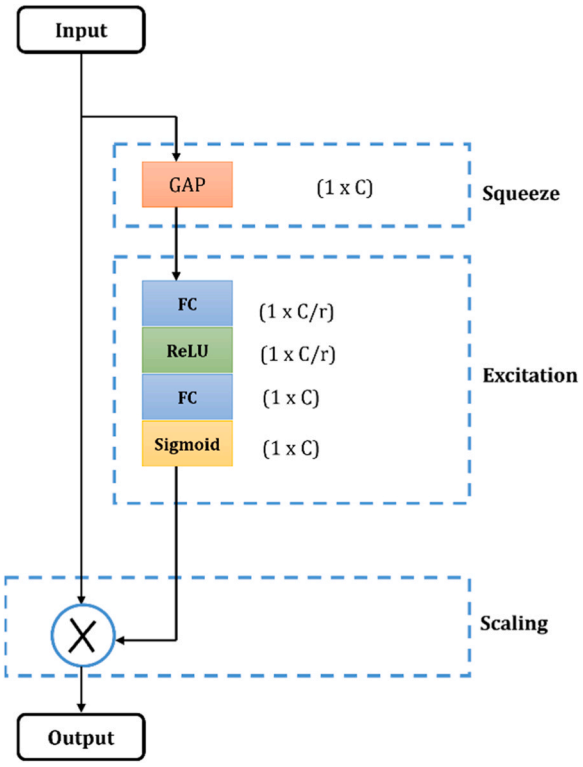

Fig. 6. Squeeze and Excitation Block.

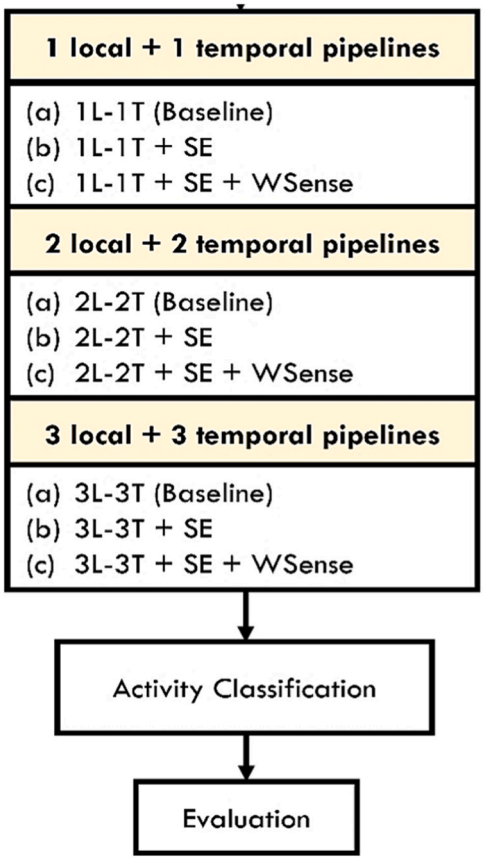

Fig. 7. Flow of experiments on DLT.

included feature weighting in the 1 L-1 T pipeline, and then the performance was evaluated before finally combining feature weighting with WSense on the 1 L-1 T pipeline. The second set of control experiments concatenated two local and two temporal (2 L-2 T) feature learning pipelines and directly classified activities, then feature weighting was included in the 2 L-2 T pipeline, and the model was evaluated before finally combining feature weighting with the WSense module on the 2 L-2 T pipeline. For the last set of experiments, three local and three temporal (3 L-3 T) feature learning pipelines were concatenated, and the pipeline was used for activity classification directly. In the second experiment, feature weighting was included in the 3 L-3 T pipeline before classifying activities. While in the final experiment, Feature weighting and WSense were combined and included in the 3S-ST feature learning pipeline before the model was evaluated. An epoch of 100 was set, and an early stopping mechanism was used in the call backs to stop the training once the model stops improving. The hyperparameters of the DLT model is shown in Table 1.

The DLT model and its control experiments were built using TensorFlow 2.7.0 with Python 3.9 and trained on a workstation equipped with RTX 3050Ti 4 GB GPU and 16 GB RAM.

4.3. Results

The DLT model uses a new approach to feature learning by combining the local and temporal features with the relationship of the sliding window. Three local and three temporal sub-pipelines, which learned features simultaneously, were concatenated.

4.3.1. Experiments on PAMAP2

The results of the 3 L-3 T feature learning pipeline experiments on PAMAP2 are presented in Table 2. The Baseline 3 L-3 T feature learning pipeline recorded a recognition accuracy of 98.25%, with eight million seven hundred and eighty-three thousand four hundred and eight four (8783,484) model parameters. The Baseline 3 L-3 T model returned 0.99 precision, recall and F1 score, while 0.98 precision and F1 with 0.97 recall were achieved on sitting activity. On walking activity, 1.00 precision with 0.99 recall and F1 was achieved, while running had 0.98 precision, 1.00 recall and 0.99 F1. Cycling activity returned 1.00 precision with 0.99 recall and F1, upstairs had 0.96 precision with 0.97 recall and F1, while downstairs had 0.95 precision with 0.96 recall and F1. Vacuum cleaning also had 0.96 precision with 0.97 recall and F1, while ironing returned 0.99 score across the three evaluation metrics, and lastly, rope jumping had a precision of 1.00, 0.94 recall, and 0.97 F1. Figs. 8and 9.

Results of the experiment on the 3 L-3 T-SE feature learning model presented in Table 2 returned a recognition accuracy of 98.45%, with eight million seven hundred and eighty-six thousand, five hundred and fifty-six (8786,556) model parameters. The classification report shows a precision, recall, and F1 of 1.00 on lying, while sitting has 0.99 precision and F1, with 0.98 recall. Standing activity returned 0.96 precision, 0.99 recall, and 0.98 F1. Walking had 0.99 precision and F1 with 1.00 recall,

Table 1
Hyperparameters on DLT Experiments.

Hyperparameters	Details
Optimizer	Adam
Epoch	100
Batch Size	PAMAP2 – 32, WISDM - 16
Learning rate	Initial Learning rate = $1e^{-4}$ Minimum Learning rate = $1e^{-7}$ Patience = 5
Model loss	Categorical cross-entropy Early stopping patience = 20
Kernel Size	5, 7, 9
Sliding window size	WISDM – 128 PAMAP2 – 171
Sliding window overlap	WISDM – 50% PAMAP2 – 50%

Table 2
Classification Report (3 L-3 T on PAMAP2).

	3 L-3 T Baseline 98.25% Model Size: 8.783 M			**3 L-3 T-SE 98.45% Model Size: 8.786 M**			**DLT 98.52% Model Size: 0.680 M**		
Activity	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Lying	0.99	0.99	0.99	1.00	1.00	1.00	1.00	0.99	0.99
Sitting	0.98	0.97	0.98	0.99	0.98	0.99	1.00	0.96	0.98
Standing	0.97	0.99	0.98	0.96	0.99	0.98	0.97	0.98	0.97
Walking	1.00	0.99	0.99	0.99	1.00	0.99	0.99	0.99	0.99
Running	0.98	1.00	0.99	1.00	1.00	1.00	1.00	0.98	0.99
Cycling	1.00	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99
Nordic walking	0.99	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99
Upstairs	0.96	0.97	0.97	0.94	0.96	0.95	0.96	1.00	0.98
Downstairs	0.95	0.96	0.96	0.96	0.96	0.96	0.99	0.96	0.98
Vacuum cleaning	0.96	0.97	0.97	0.98	0.96	0.97	0.97	0.98	0.97
Ironing	0.99	0.99	0.99	0.99	0.98	0.98	0.99	0.99	0.99
Rope jumping	1.00	0.94	0.97	1.00	0.98	0.99	0.98	0.98	0.98

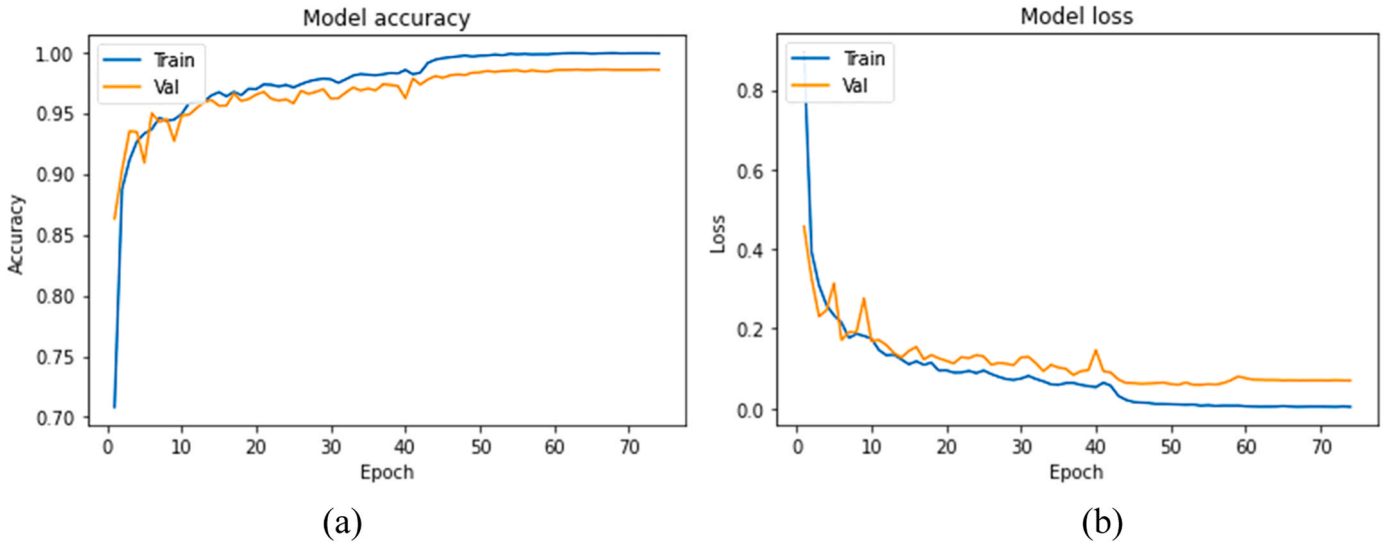

Fig. 8. (a) Model training and validation on PAMAP2(a) accuracy (b) loss.

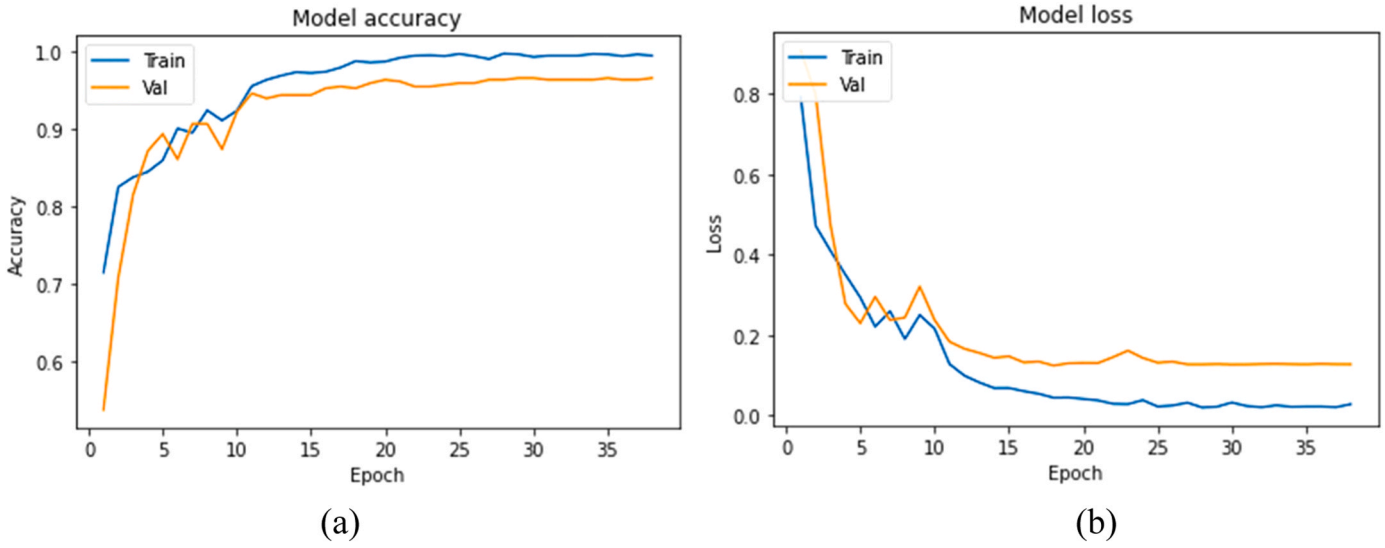

Fig. 9. (a) Model training and validation on WISDM (a) accuracy (b) loss.

running activity achieved 1.00 score across the three evaluation metrics, while cycling had 0.99 precision with 1.00 recall and F1. On Nordic walking activity, a precision of 1.00 was returned with 0.99 recall and F1, while walking upstairs had 0.94 precision, 0.96 recall, and 0.95 F1, walking downstairs had 0.96 across the three evaluation metrics. The report on vacuum cleaning activity showed a precision of 0.98, 0.96 recall, and 0.97 F1. Ironing had 0.99 precision with 0.98 recall and F1, while rope jumping had 1.0 precision, 0.98 recall and 0.99 F1.

As shown in Table 3, the experiment on the 3 L-3 T-SE-WSense (DLT) feature learning model on PAMAP2 returned a recognition accuracy of 98.52% with six hundred and eighty thousand nine hundred and seventy-two (680,972) model parameters. The classification report shows that the 3 L-3 T-SE-WSense model achieved a precision of 1.00 on lying activity, with 0.99 recall and F1 score. On sitting activity, 1.00 precision was also returned with 0.96 recall and 0.98 F1. Standing activity had 0.98 recall with 0.97 precision and F1, walking activity had 0.99 score across the three evaluation metrics, running had a precision score of 1.00, recall of 0.98, and F1 of 0.99. Report on cycling, Nordic walking, and ironing activities returned 0.99 score across the three-evaluation metrics, walking upstairs had 0.96 precision, 1.00 recall, and 0.98 F1 score, while walking downstairs had 0.99 precision, 0.96 recall, and 0.98 F1. Report on vacuum cleaning activity returned 0.98 recall with 0.97 precision and F1, and rope jumping had 0.98 score across the three-evaluation metrics. The confusion matrix of the DLT model is presented in Table 3.

The confusion matrix shown in Table 3 shows that the 3 L-3 T-SE-WSense model correctly classified 455 lying samples, with 1 misclassified as sitting and 2 as ascending stairs. 425 samples of sitting were correctly classified, and 2 samples were misclassified as lying, and 11 as standing, while 2 samples were misclassified as vacuum cleaning. On standing activity, 440 samples were correctly classified, and 1 sample was misclassified as cycling, 4 as vacuum cleaning and 6 as ironing. 515 samples of walking activity were correctly classified, with 2 samples misclassified as standing, 2 as ascending stairs, and 2 as vacuum cleaning. On running activity, 215 samples were correctly classified, and 1 sample was misclassified as walking, 1 as ascending stairs, and 2 as vacuum cleaning.

Cycling activity had a total of 384 samples, and 382 were correctly classified, while 1 sample was misclassified as Nordic walking, and another as vacuum cleaning. On the Nordic walking activity, 398 samples were correctly classified with 2 misclassified as walking, and 1 as vacuum cleaning. 242 samples of ascending stairs were correctly classified with 1 sample misclassified as descending stairs. Also, 218 descending stairs activities were correctly classified, while 5 samples were misclassified as ascending stairs, 2 as walking and 1 as vacuum cleaning. 427 vacuum cleaning activities were correctly classified, with 1 sample misclassified sitting, 2 as cycling, 1 as Nordic walking, 1 as ascending stairs and another two samples were misclassified as ironing. Ironing activity had 586 samples which were correctly classified, with 1 sample misclassified as standing, 1 as vacuum cleaning, and another as rope jumping. Lastly, out of the total 111 samples of rope jumping, 109 were correctly classified, while 1 sample was misclassified as descending stairs and another sample as ascending stairs.

4.3.2. Experiments on WISDM

The results of the 3 L-3 T feature learning pipeline experiments on WISDM are presented in Table 4. As shown in Table 4, the Baseline 3 L-3 T model recorded a recognition accuracy of 96.85% with twelve million two hundred and eighty-five thousand two hundred and twenty (12,285,222) parameters. The classification report of the Baseline 3 L-3 T model showed that walking downstairs activity had a precision of 0.83, 0.89 recall, and F1 of 0.86. On jogging activity, a precision of 0.99 was achieved with 1.00 recall and F1. Sitting had a precision of 0.89, 1.00 recall, and 0.94 F1. On standing activity, a 1.00 precision was achieved, with 0.83 recall, and 0.91 F1. Walking upstairs had a precision of 0.89, 0.81 recall, and 0.85 F1. While walking activity had a 1.00 score across the three metrics.

Results on the 3 L-3 T-SE model achieved recognition accuracy of 97.55% with twelve million two hundred and eighty-eight thousand two hundred and ninety-four (12,288,294) parameters. The classification report of 3 L-3 T-SE model presented in Table 4 shows that walking downstairs had a precision of 0.89, 0.87 recall, and 0.88 F1. Jogging and walking activities recorded 1.00 score across the three evaluation metrics, while sitting had 0.89 precision, 1.00 recall, and 0.94 F1. Standing had 1.00 precision, 0.83 recall and 0.91 F1, while walking upstairs activity had a precision of 0.89, recall of 0.92 and 0.90 F1.

The DLT model achieved a recognition accuracy of 97.90%, with six hundred and fifty-five thousand nine hundred and ten (655,910) parameters. The classification report shows that a precision of 0.89 was recorded on walking downstairs with 0.91 recall, and 0.90 F1. Jogging, standing, sitting and walking had 1.00 scores across the three-evaluation metrics, while walking upstairs recorded a 0.91 precision, 0.90 recall and 0.91 F1, showing that the proposed DLT model extracted improved features compared to the baselines. The confusion matrix of the DLT presented in Table 5, shows that out of the 55 walking downstairs samples used for model testing, 50 samples were correctly classified, with 5 misclassified as walking upstairs. On jogging activity, 1 sample out of the total 215 samples was misclassified as walking downstairs, while the remaining 214 were correctly classified. Walking upstairs activity, which had 59 test samples, had 53 correctly classified samples, with 5 samples misclassified as walking downstairs and 1 sample misclassified as walking. On sitting and standing activities, 8 and 6 samples were correctly classified, respectively, the total samples used for model testing, while walking had 229 of its samples correctly classified.

4.4. Ablation study

Ablation studies were carried out to determine the batch size and the number of neurons in the Bi-LSTM layer, and the results are presented in Fig. 10 (a) and (b). Batch sizes 8, 16, 32, 64 and 128 were considered in the ablation study. As shown in Fig. 10(a), batch size 32 achieved the highest recognition accuracy on PAMAP2 dataset. Also, 16, 32, 64, 128,

Table 3
Confusion Matrix of DLT on PAMAP2.

Activity	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	455	1	0	0	0	0	0	2	0	0	0	0
A2	2	425	11	0	0	0	0	0	0	2	0	0
A3	0	0	440	0	0	1	0	0	0	4	6	0
A4	0	0	2	515	0	0	0	2	0	2	0	0
A5	0	0	0	1	215	0	0	1	0	2	0	0
A6	0	0	0	0	0	382	1	0	0	1	0	0
A7	0	0	0	2	0	0	398	0	0	1	0	0
A8	0	0	0	0	0	0	0	242	1	0	0	0
A9	0	0	0	2	0	0	0	5	218	1	0	0
A10	0	1	0	0	0	2	1	1	0	427	2	0
A11	0	0	1	0	0	0	0	0	0	1	586	1
A12	0	0	0	0	0	0	0	0	1	1	0	109

Table 4
Classification Report (3 L-3 T on WISDM).

	Baseline 3 L-3 T 96.85% Model Size: 12.285 M			3 L-3 T-SE 97.55% Model Size: 12.288 M			DLT 97.90% Model Size: 0.655 M		
Activity	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Downstairs	0.83	0.89	0.86	0.89	0.87	0.88	0.89	0.91	0.90
Jogging	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sitting	0.89	1.00	0.94	0.89	1.00	0.94	1.00	1.00	1.00
Standing	1.00	0.83	0.91	1.00	0.83	0.91	1.00	1.00	1.00
Upstairs	0.89	0.81	0.85	0.89	0.92	0.90	0.91	0.90	0.91
Walking	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 5
Confusion Matrix of DLT on WISDM.

Activity	Downstairs	Jogging	Sitting	Standing	Upstairs	Walking
Downstairs	50	0	0	0	5	0
Jogging	1	214	0	0	0	0
Sitting	0	0	8	0	0	0
Standing	0	0	0	6	0	0
Upstairs	5	0	0	0	53	1
Walking	0	0	0	0	0	229

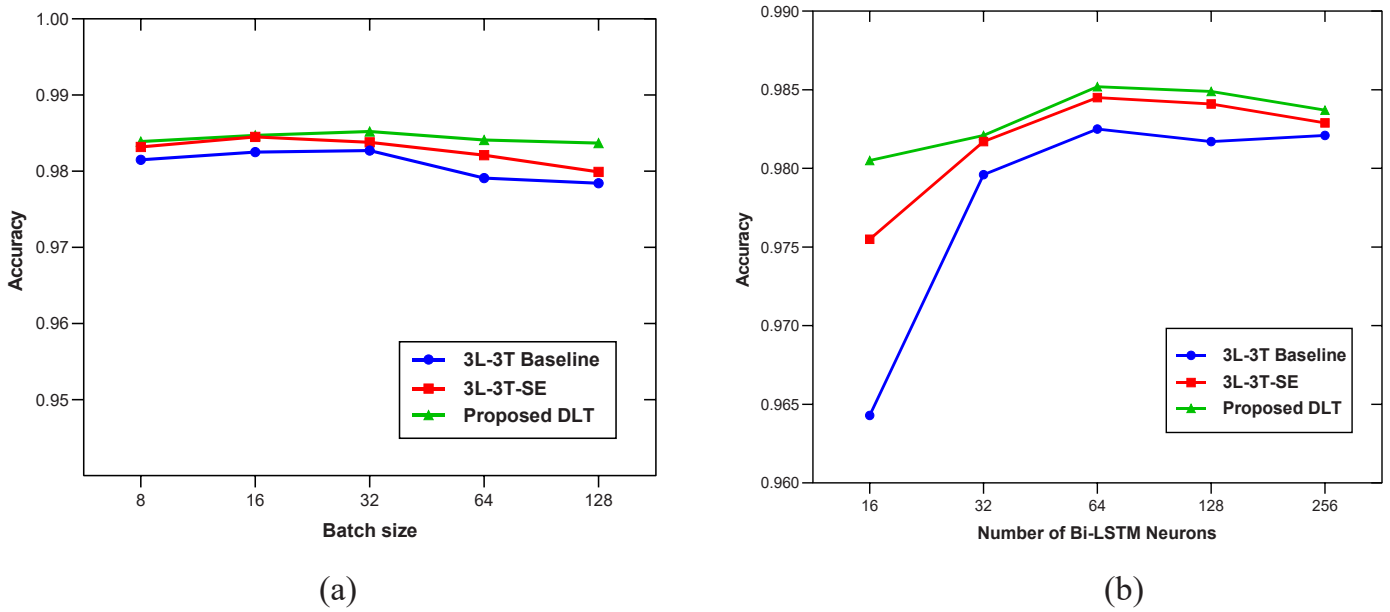

Fig. 10. (a) Batch size comparison on PAMAP2 (b) Comparison of the number of neurons in the Bi-LSTM layer on PAMAP2.

and 256 neurons were considered in the experiment. However, the highest recognition performance was achieved when 64 neurons were used in the Bi-LSTM layer of the proposed DLT model, as shown in Fig. 10(b). Similarly, the results of the ablation study on the WISDM dataset, presented in Fig. 11 (a) and (b), showed that batch size 16 returned the highest recognition performance, and this was achieved using 64 neurons in the Bi-LSTM layer. As shown in Fig. 11(b), 256 neurons in the Bi-LSTM layer returned the highest recognition accuracy on the 3 L-3 T-SE baseline. However, the result achieved when 64 neurons were used in the Bi-LSTM layer was presented for fair comparison.

4.4.1. *Experiments with pipelines*

As shown in Table 6, when one local sub-pipeline was concatenated with one temporal sub-pipeline (1 L-1 T), a recognition accuracy of 95.62% was achieved with 4.271 million parameters on the WISDM dataset. Including the SE block in the 1 L-1 T sub-pipelines improved the recognition accuracy to 96.85%, with additional parameters of 4.272 million. However, when the WSense module was added to the concatenated sub-pipelines, the recognition accuracy improved to 97.02%, and the model parameters was reduced to 0.569 million. Also, the result of the 1 L-1 T baseline model on the PAMAP2 dataset presented in Table 6 returned an accuracy of 96.92% with 3.104 million parameters. Including the SE block into the 1 L-1 T pipeline improved the performance to 97.20%, with 3.105 million parameters. Likewise, when the WSense module was added to the 1 L-1 T sub-pipelines, the recognition accuracy increased to 97.27% with 0.582 million parameters. By combining 2 local and 2 temporal pipelines (2 L-2 T), results on the WISDM dataset show that the Baseline 2 L-2 T model achieved 96.67% accuracy, an improvement on the Baseline 1 L-1 T. However, the model has a size of 8.278 million parameters. Also, when the SE block was included in the 2 L-2 T pipeline, the model saw an increase in

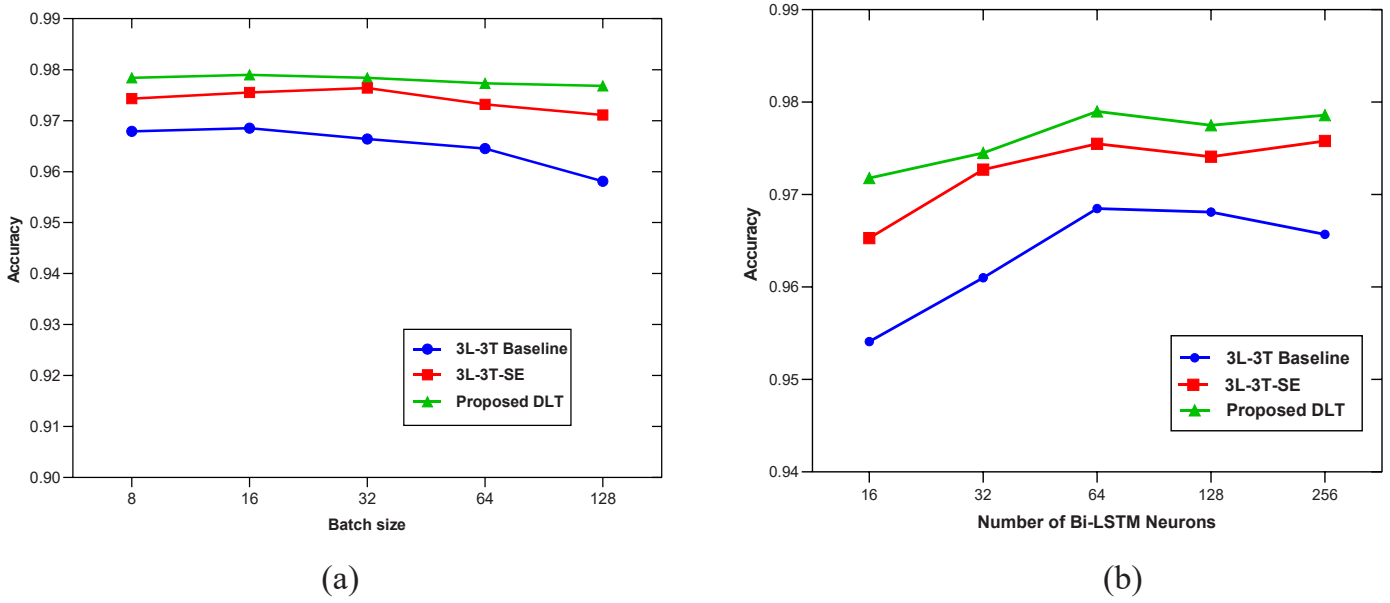

Fig. 11. (a) Batch size comparison on WISDM (b) Comparison of the number of neurons in the Bi-LSTM layer on WISDM.

Table 6
Experiments with Pipelines on WISDM and PAMAP2.

	Pipelines	Accuracy (%)	Model Size
WISDM	1 L-1 T feature learning sub-pipelines	95.62	4.271 M
	2 L-2 T feature learning sub-pipelines	96.67	8.278 M
	3 L-3 T feature learning sub-pipelines	96.85	12.285 M
	1 L-1 T-SE feature learning sub-pipelines	96.85	4.272 M
	2 L-2 T-SE feature learning sub-pipelines	97.02	8.280 M
	3 L-3 T-SE feature learning sub-pipelines	97.55	12.288 M
	1 L-1 T-SE-WSense	97.02	**0.569 M**
	2 L-2 T-SE-WSense	97.37	0.580 M
	Proposed DLT	**97.90**	0.655 M
PAMAP2	1 L-1 T feature learning sub-pipelines	96.92	3.104 M
	2 L-2 T feature learning sub-pipelines	97.96	5.944 M
	3 L-3 T feature learning sub-pipelines	98.25	8.783 M
	1 L-1 T-SE feature learning sub-pipelines	97.20	3.105 M
	2 L-2 T-SE feature learning sub-pipelines	98.28	5.946 M
	3 L-3 T-SE feature learning sub-pipelines	98.45	8.786 M
	1 L-1 T-SE-WSense	97.27	**0.517 M**
	2 L-2 T-SE-WSense	98.36	0.605 M
	Proposed DLT	**98.52**	0.680 M

recognition accuracy, as the accuracy stood at 97.02%, but also with a high model parameter of 8.280 million parameters. However, plugging in the WSense module on the 2 L-2 T-SE pipeline reduced the model size to 0.645 million, and the recognition accuracy improved to 97.37%.

Similarly, on the PAMAP2 dataset, the 2 L-2 T baseline model recorded an accuracy of 97.76% with 5.944 million parameters, as shown in Table 6. When the SE block was added, the recognition accuracy improved to 98.28%, with 5.946 million parameters. However, the high model size was reduced to 0.670 million parameters when the WSense module was plugged into the 2 L-2 T-SE feature learning pipeline, and the recognition accuracy also increased to 98.36%. On the 3 L-3 T model, which concatenated three local feature learning pipelines with three temporal pipelines simultaneously, a recognition accuracy of 96.85% was achieved on the WISDM dataset, with 12.285 million parameters. Likewise, when the SE block was added to the 3 L-3 T model, the recognition accuracy improved to 97.55%, with 12.288 million parameters. However, the DLT model achieved a state-of-the-art recognition accuracy of 97.90%, with 0.655 million parameters.

Likewise, on the PAMAP2 dataset, the 3 L-3 T baseline model achieved a recognition accuracy of 98.25% with 8.783 million parameters as shown in Table 6, while the 3 L-3 T-SE model improved the result by achieving an accuracy of 98.45% with 8.786 million parameters. However, the DLT model reduced the model parameters to 0.680 million and achieved a state-of-the-art recognition accuracy of 98.52%. A comparison of the pipelines with accuracy and model size on the two datasets is presented in Fig. 12 and Fig. 13.

4.5. Comparison with state-of-the-art

The comparison of the proposed DLT architecture with current state-of-the-art models in terms of methodology, model size, and accuracy is presented in Table 7. As shown, Gao et al. [20] developed a dual attention model and achieved a recognition accuracy of 93.16% on the PAMAP2 dataset with 3.51 M parameters. Similarly, for enhanced feature learning from activity signals, Dua et al. [48] suggested a CNN and GRU model with multiple inputs and achieved recognition accuracy of 95.24% on PAMAP2 and 97.21% on the WISDM dataset. Even though the size of the model was not presented in the research, the stacking structure of the layers shows that the size of the model will be bulky, as a fully connected layer was connected to the concatenation of the three-feature learning pipeline after two GRU layers, with no mechanism to reduce the size.

Also, in Challa et al. [54], another multiple input model was proposed with CNN and Bi-LSTM and achieved recognition accuracy of 94.29% and 96.05% with 0.647 M and 0.622 M parameters on PAMAP2 and WISDM datasets, respectively. Likewise, in Han et al. [69], a heterogenous CNN module was proposed to improve feature learning in activity recognition and achieved an accuracy of 92.97% on PAMAP2 with 1.37 M parameters. A similar deep learning model was proposed by Xiao et al. [70] to encode local and temporal information of the input data and achieved an F-Score of 98.00%. The model's size was not presented, but the two-stream feature learning pipelines suggest a large number of parameters. Bhattacharya et al. [58] proposed an ensemble of CNN, CNN-LSTM, LSTM, and other models and evaluated on several datasets. However, replication showed that the model is parameter-heavy. Even though these models achieved improved

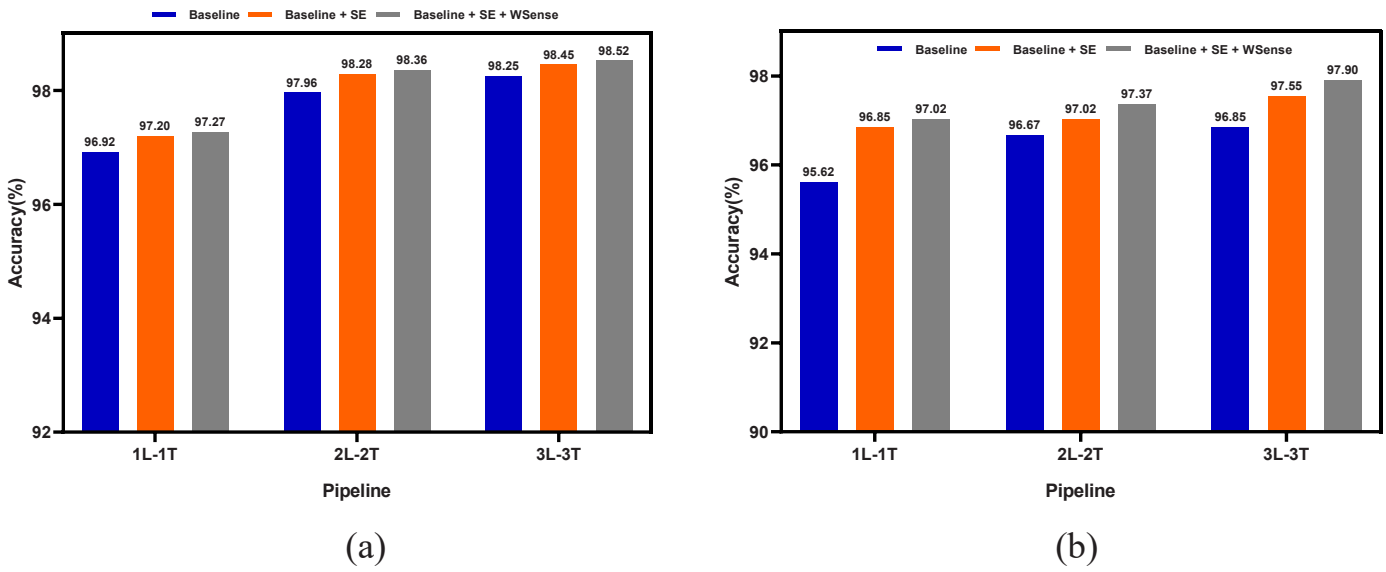

Fig. 12. Comparison of Accuracy and Pipelines (a) PAMAP2 (b) WISDM.

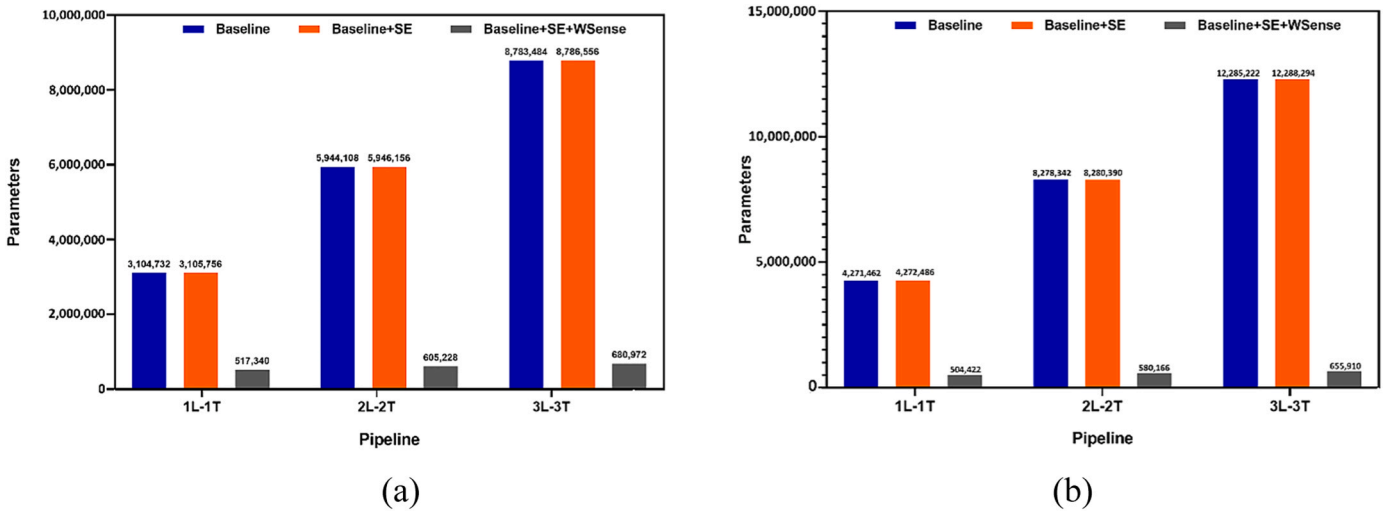

Fig. 13. Comparison of Model size and Pipeline (a) PAMAP2 (b) WISDM.

performance, the limitation synonymous with them is the recognition accuracy recorded and the bulky size of the models, which is a constraint when deploying activity recognition models on portable devices. However, with the proposed DLT architecture, a state-of-the-art accuracy of 98.52% was recorded on PAMAP2, while 97.90% was achieved on the WISDM dataset, which outperformed recent models, and this was achieved using a lightweight architecture.

5. Conclusion

Identifying human activities from wearable sensor signals is a challenging task that calls for contributions from researchers. In order to improve feature learning from wearable sensors, several multi-input architectures have been proposed. However, these architectures often extract local and temporal features on a pipeline, affecting the feature representation quality. Also, such models are parameter-heavy due to the number of weights involved in the architecture. Since resources (CPU, battery, and memory) of end devices are limited, it is important to propose lightweight deep architectures for easy deployment on end devices. In this paper, we, for the first time, propose a new method of feature learning by extracting local features in the current windows on a different sub-pipeline and temporal features on other sub-pipelines simultaneously. Then, the features were concatenated before using channel attention to improve responsiveness to discriminative features. By leveraging this approach, we were able to take advantage of the capabilities of CNNs and RNNs fully for feature learning in HAR. The proposed method, called DLT, was validated on WISDM and PAMAP2 datasets, and the results showed that the DLT was able to improve feature learning, compared to the existing methods. In order to determine the suitable number of pipelines for the DLT architecture, several experiments were carried out using 1 Local - 1 Temporal, 2 Local – 2 Temporal, and 3 Local – 3 Temporal feature learning sub-pipelines. The 98.52% achieved by the DLT model on PAMAP2 is currently state-of-the-art, while the 97.90% achieved on WISDM outperformed several existing feature learning architectures, and this was achieved using a few model parameters. This makes the DLT a deep, lightweight human activity recognition model that can be deployed on end devices for activity monitoring across various domains. For future work, we plan to infuse attention mechanisms into each local feature learning sub-pipeline and transformers for temporal feature learning to improve the quality of features extracted to infer activities. Also, more sensor-rich datasets, including datasets with transitional activities, will be considered.

Table 7
Comparison with State-of-the-art models.

Author	Year	Method	Accuracy	Parameters
Gil-Martín et al. [68]	2021	Sub-Window CNN	PAMAP2: 97.22%	3.701 M
Gao et al. [20]	2021	DanHAR	PAMAP2: 93.16% WISDM: 98.85%	3.51 M 2.33 M
Dua et al. [48]	2021	Mult-input CNN-GRU	PAMAP2 95.27% WISDM 97.21%	- -
Challa et al., [54]	2021	Multibranch CNN-BiLSTM	PAMAP2 94.29% WISDM 96.05%	0.647 M 0.622 M
Lu et al. [57]	2022	Multi-channel CNN-GRU	PAMAP: 96.25% WISDM: 96.41%	-
Han et al. [69]	2022	Heterogeneous CNN	PAMAP2 92.97%	1.37 M
Bhattacharya et al. [58]	2022	Ensem-HAR	PAMAP: 97.45% WISDM: 98.70%	6.45 M 5.68 M
Mim et al. [8]	2023	GRU-INC	PAMAP: 95.61%	0.723 M
Proposed Model		DLT	**PAMAP2: 98.52%** WISDM: 97.90%	**0.680 M** **0.655 M**

CRediT authorship contribution statement

Ayokunle Olalekan Ige: Conceptualization, Methodology, Software, Writing – original draft preparation. **Mohd Halim Mohd Noor**: Supervision, Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

This study uses public datasets.

References

[1] WHO, WHO j Ageing and life-course, 2023. https://www.who.int/health-topics/ageing (accessed February 27, 2023).
[2] M. Webber, R.F. Rojas, Human activity recognition with accelerometer and gyroscope: a data fusion approach, IEEE Sens. J. 21 (2021) 16979–16989, https://doi.org/10.1109/JSEN.2021.3079883.
[3] O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors, IEEE Commun. Surv. Tutor. (2013) 1192–1209, https://doi.org/10.1029/GL002i002p00063.
[4] A.O. Ige, M.H. Mohd Noor, A survey on unsupervised learning for wearable sensor-based activity recognition, Appl. Soft Comput. (2022), 109363, https://doi.org/10.1016/j.asoc.2022.109363.
[5] M. Abdel-Basset, H. Hawash, R.K. Chakrabortty, M. Ryan, M. Elhoseny, H. Song, ST-DeepHAR: Deep Learning Model for Human Activity Recognition in IoHT Applications, IEEE Internet Things J. 8 (2021) 4969–4979, https://doi.org/10.1109/JIOT.2020.3033430.
[6] M.H. Mohd Noor, Feature learning using convolutional denoising autoencoder for activity recognition, Neural Comput. Appl. 33 (2021) 10909–10922, https://doi.org/10.1007/s00521-020-05638-4.
[7] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, Y. Liu, Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities, ACM Comput. Surv. 54 (2021) 1–40, https://doi.org/10.1145/3447744.
[8] T.R. Mim, M. Amatullah, S. Afreen, M.A. Yousuf, S. Uddin, S.A. Alyami, K.F. Hasan, M.A. Moni, GRU-INC: An inception-attention based approach using GRU for human activity recognition, Expert Syst. Appl. 216 (2023), 119419, https://doi.org/10.1016/j.eswa.2022.119419.
[9] F.M. Rueda, R. Grzeszick, G.A. Fink, S. Feldhorst, M. Ten Hompel, Convolutional neural networks for human activity recognition using body-worn sensors, Informatics 5 (2018) 1–17, https://doi.org/10.3390/informatics5020026.
[10] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, A. Aliverti, A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone, Sens. Switz. 19 (2019), https://doi.org/10.3390/s19173731.
[11] L. Bai, L. Yao, X. Wang, S.S. Kanhere, Y. Xiao, Prototype similarity learning for activity recognition, Pac. -Asia Conf. Knowl. Discov. Data Min. (2020) 649–661.
[12] Y. Chen, K. Zhong, J. Zhang, Q. Sun, X. Zhao, LSTM Networks for Mobile Human Activity Recognition, : Int. Conf. Artif. Intell. Technol. Appl. (2016) 50–53, https://doi.org/10.2991/icaita-16.2016.13.
[13] Y. Guan, T. Plötz, Ensembles of deep LSTM learners for activity recognition using wearables, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1 (2017) 1–28, https://doi.org/10.1145/3090076.
[14] S.S. Saha, S.S. Sandha, M. Srivastava, Deep Convolutional Bidirectional LSTM for Complex Activity Recognition with Missing Data, Springer, Singapore, 2021, https://doi.org/10.1007/978-981-15-8269-1_4.
[15] J. Donahue, L.A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 677–691, https://doi.org/10.1109/TPAMI.2016.2599174.
[16] K. Xia, J. Huang, H. Wang, LSTM-CNN architecture for human activity recognition, IEEE Access 8 (2020) 56855–56866, https://doi.org/10.1109/ACCESS.2020.2982225.
[17] M.H. Mohd Noor, S.Y. Tan, M.N. Ab Wahab, Deep Temporal Conv-LSTM for Activity Recognition, Neural Process. Lett. (2022), https://doi.org/10.1007/s11063-022-10799-5.
[18] H. Park, N. Kim, G.H. Lee, J.K. Choi, MultiCNN-FilterLSTM: Resource-efficient sensor-based human activity recognition in IoT applications, Future Gener. Comput. Syst. 139 (2023) 196–209, https://doi.org/10.1016/j.future.2022.09.024.
[19] A.O. Ige, M.H. Mohd Noor, Unsupervised feature learning in activity recognition using convolutional denoising autoencoders with squeeze and excitation networks, ICOIACT 2022 - 5th Int. Conf. Inf. Commun. Technol. N. Way Make AI Useful Everyone N. Norm. Era Proc. (2022) 435–440, https://doi.org/10.1109/ICOIACT55506.2022.9972095.
[20] W. Gao, L. Zhang, Q. Teng, J. He, H. Wu, DanHAR: Dual Attention Network for multimodal human activity recognition using wearable sensors, Appl. Soft Comput. 111 (2021), 107728, https://doi.org/10.1016/j.asoc.2021.107728.
[21] Z.N. Khan, J. Ahmad, Attention induced multi-head convolutional neural network for human activity recognition, Appl. Soft Comput. 110 (2021), 107671, https://doi.org/10.1016/j.asoc.2021.107671.
[22] H. Ma, W. Li, X. Zhang, S. Gao, S. Lu, Attnsense: Multi-level attention mechanism for multimodal human activity recognition, IJCAI Int. Jt. Conf. Artif. Intell. 2019-Augus (2019) 3109–3115, https://doi.org/10.24963/ijcai.2019/431.
[23] E. Essa, I.R. Abdelmaksoud, Temporal-channel convolution with self-attention network for human activity recognition using wearable sensors, Knowl. -Based Syst. 278 (2023), 110867, https://doi.org/10.1016/j.knosys.2023.110867.
[24] Y. Zhou, H. Zhao, Y. Huang, M. Hefenbrock, T. Riedel, M. Beigl, TinyHAR: A Lightweight Deep Learning Model Designed for Human Activity Recognition, Assoc. Comput. Mach. (2022), https://doi.org/10.1145/3544794.3558467.
[25] S. Bhattacharya, P. Nurmi, N. Hammerla, T. Plötz, Using unlabeled data in a sparse-coding framework for human activity recognition, Pervasive Mob. Comput. 15 (2014) 242–262, https://doi.org/10.1016/j.pmcj.2014.05.006.
[26] A.R. Javed, R. Faheem, M. Asim, T. Baker, M.O. Beg, A smartphone sensors-based personalized human activity recognition system for sustainable smart cities, Sustain. Cities Soc. 71 (2021), 102970, https://doi.org/10.1016/j.scs.2021.102970.
[27] M.G. Rasul, M.H. Khan, L.N. Lota, Nurse care activity recognition based on convolution neural network for accelerometer data. UbiCompISWC 2020 Adjun. - Proc. 2020 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Proc. 2020 ACM Int. Symp. Wearable Comput, 2020, pp. 425–430, https://doi.org/10.1145/3410530.3414335.
[28] M. Babiker, O.O. Khalifa, K.K. Htike, A. Hassan, M. Zaharadeen, Automated daily human activity recognition for video surveillance using neural network, 2017 IEEE Int. Conf. Smart Instrum. Meas. Appl. ICSIMA 2018 (2017) 1–5, https://doi.org/10.1109/ICSIMA.2017.8312024.
[29] K. Mitsis, K. Zarkogianni, E. Kalafatis, K. Dalakleidi, A. Jaafar, G. Mourkousis, K. S. Nikita, A multimodal approach for real time recognition of engagement towards adaptive serious games for health, Sensors 22 (2022), https://doi.org/10.3390/s22072472.
[30] S. Khare, S. Sarkar, M. Totaro, Comparison of sensor-based datasets for human activity recognition in wearable IoT. IEEE World Forum Internet Things WF-IoT 2020 - Symp. Proc, 2020, pp. 1–6, https://doi.org/10.1109/WF-IoT48130.2020.9221408.
[31] C. Wang, Y. Gao, A. Mathur, A.C. Amanda, N.D. Lane, N. Bianchi-Berthouze, Leveraging activity recognition to enable protective behavior detection in continuous data, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 5 (2021) 1–24, https://doi.org/10.1145/3463508.
[32] J. Liu, Convolutional neural network-based human movement recognition algorithm in sports analysis, Front. Psychol. 12 (2021) 1738, https://doi.org/10.3389/fpsyg.2021.663359.

[33] J. Manjarres, P. Narvaez, K. Gasser, W. Percybrooks, M. Pardo, Physical workload tracking using human activity recognition with wearable devices, Sens. Switz. 20 (2020) 39, https://doi.org/10.3390/s20010039.
[34] M.H.M. Noor, A. Nazir, M.N.A. Wahab, J.O.Y. Ling, Detection of freezing of gait using unsupervised convolutional denoising autoencoder, IEEE Access 9 (2021) 115700–115709, https://doi.org/10.1109/ACCESS.2021.3104975.
[35] S. Wang, G. Zhou, A review on radio based activity recognition, Digit. Commun. Netw. 1 (2015) 20–29, https://doi.org/10.1016/j.dcan.2015.02.006.
[36] W. Qi, H. Su, F. Chen, X. Zhou, Y. Shi, G. Ferrigno, E. De Momi, Depth vision guided human activity recognition in surgical procedure using wearable multisensor. ICARM 2020 - 2020 5th IEEE Int. Conf. Adv. Robot. Mechatron, 2020, pp. 431–436, https://doi.org/10.1109/ICARM49381.2020.9195356.
[37] S.K. Yadav, K. Tiwari, H.M. Pandey, S.A. Akbar, A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions, Knowl. -Based Syst. 223 (2021), 106970, https://doi.org/10.1016/j.knosys.2021.106970.
[38] F. Demrozi, G. Pravadelli, A. Bihorac, P. Rashidi, Human activity recognition using inertial, physiological and environmental sensors: a comprehensive survey, IEEE Access 8 (2020) 210816–210836, https://doi.org/10.1109/ACCESS.2020.3037715.
[39] A. Ferrari, D. Micucci, M. Mobilio, P. Napoletano, Hand-crafted Features vs Residual Networks for Human Activities Recognition using accelerometer, in: 2019 IEEE 23rd Int. Symp. Consum. Technol, 2019, ISCT, 2019, pp. 153–156, https://doi.org/10.1109/ISCE.2019.8901021.
[40] S. Sani, N. Wiratunga, S. Massie, K. Cooper, kNN sampling for personalised human activity recognition, Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma. (2017) 330–344, https://doi.org/10.1007/978-3-319-61030-6_23.
[41] K.G. Manosha Chathuramali, R. Rodrigo, Faster human activity recognition with SVM, Int. Conf. Adv. ICT Emerg. Reg. ICTer 2012 - Conf. Proc. (2012) 197–203, https://doi.org/10.1109/ICTer.2012.6421415.
[42] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, https://doi.org/10.1038/nature14539.
[43] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, J. Zhang, Convolutional Neural Networks for human activity recognition using mobile sensors. Proc. 2014 6th Int. Conf. Mob. Comput. Appl. Serv, MobiCASE 2014,, New York, NY, USA, 2014, pp. 197–205, https://doi.org/10.4108/icst.mobicase.2014.257786.
[44] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J.L. Zhao, Time series classification using multi-channels deep convolutional neural networks, Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma. 8485 LNCS (2014) 298–310, https://doi.org/10.1007/978-3-319-08010-9_33.
[45] C.A. Ronao, S.B. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, Expert Syst. Appl. 59 (2016) 235–244, https://doi.org/10.1016/j.eswa.2016.04.032.
[46] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, J. Zhou, TSE-CNN: A Two-Stage End-to-End CNN for Human Activity Recognition, IEEE J. Biomed. Health Inform. 24 (2020) 292–299, https://doi.org/10.1109/JBHI.2019.2909688.
[47] Z. Ahmad, N. Khan, CNN-Based Multistage Gated Average Fusion (MGAF) for Human Action Recognition Using Depth and Inertial Sensors, IEEE Sens. J. 21 (2021) 3623–3634, https://doi.org/10.1109/JSEN.2020.3028561.
[48] N. Dua, S.N. Singh, V.B. Semwal, Multi-input CNN-GRU based human activity recognition using wearable sensors, Computing 103 (2021) 1461–1478, https://doi.org/10.1007/s00607-021-00928-8.
[49] P. Agarwal, M. Alam, A lightweight deep learning model for human activity recognition on edge devices, Procedia Comput. Sci. 167 (2020) 2364–2373, https://doi.org/10.1016/j.procs.2020.03.289.
[50] M. Edel, E. Köppe, Binarized-BLSTM-RNN based Human Activity Recognition, 2016 Int. Conf. Indoor Position. Indoor Navig. IPIN 2016 (2016) 4–7, https://doi.org/10.1109/IPIN.2016.7743581.
[51] O. Barut, L. Zhou, Y. Luo, Multitask LSTM model for human activity recognition and intensity estimation using wearable sensor data, IEEE Internet Things J. 7 (2020) 8760–8768, https://doi.org/10.1109/JIOT.2020.2996578.
[52] C. Xu, D. Chai, J. He, X. Zhang, S. Duan, InnoHAR: A deep neural network for complex human activity recognition, IEEE Access 7 (2019) 9893–9902, https://doi.org/10.1109/ACCESS.2018.2890675.
[53] A. Gumaei, M.M. Hassan, A. Alelaiwi, H. Alsalman, A hybrid deep learning model for human activity recognition using multimodal body sensing data, IEEE Access 7 (2019) 99152–99160, https://doi.org/10.1109/ACCESS.2019.2927134.
[54] S.K. Challa, A. Kumar, V.B. Semwal, A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data, Vis. Comput. (2021), https://doi.org/10.1007/s00371-021-02283-3.
[55] O. Nafea, W. Abdul, G. Muhammad, M. Alsulaiman, Sensor-based human activity recognition with spatio-temporal deep learning, Sensors 21 (2021) 1–20, https://doi.org/10.3390/s21062141.
[56] Y. Li, L. Wang, Human activity recognition based on residual network and BiLSTM, Sensors 22 (2022) 1–18, https://doi.org/10.3390/s22020635.
[57] L. Lu, C. Zhang, K. Cao, T. Deng, Q. Yang, A Multi-channel CNN-GRU Model for Human Activity Recognition, IEEE Access 10 (2022) 66797–66810, https://doi.org/10.1109/ACCESS.2022.3185112.
[58] D. Bhattacharya, D. Sharma, W. Kim, M.F. Ijaz, P.K. Singh, Ensem-HAR: An Ensemble Deep Learning Model for Smartphone Sensor-Based Human Activity Recognition for Measurement of Elderly Health Monitoring, Biosensors 12 (2022), https://doi.org/10.3390/bios12060393.
[59] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, 2018, pp. 7132–7141, https://doi.org/10.1109/CVPR.2018.00745.
[60] V.S. Murahari, T. Plotz, On attention models for human activity recognition, Proc. - Int. Symp. Wearable Comput. Iswc. (2018) 100–103, https://doi.org/10.1145/3267242.3267287.
[61] H. Zhang, Z. Xiao, J. Wang, F. Li, E. Szczerbicki, A Novel IoT-Perceptive Human Activity Recognition (HAR) Approach Using Multihead Convolutional Attention, IEEE Internet Things J. 7 (2020) 1072–1080, https://doi.org/10.1109/JIOT.2019.2949715.
[62] W. Zhang, T. Zhu, C. Yang, J. Xiao, H. Ning, Sensors-based Human Activity Recognition with Convolutional Neural Network and Attention Mechanism. Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci, ICSESS,, 2020, pp. 158–162, https://doi.org/10.1109/ICSESS49938.2020.9237720.
[63] A.O. Ige, M.H. Mohd Noor, A lightweight deep learning with feature weighting for activity recognition, Comput. Intell. 39 (2023) 315–343, https://doi.org/10.1111/coin.12565.
[64] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, B. Zhao, A federated learning system with enhanced feature extraction for human activity recognition, Knowl. -Based Syst. 229 (2021), 107338, https://doi.org/10.1016/j.knosys.2021.107338.
[65] A.O. Ige, M.H.M. Noor, WSense: a robust feature learning module for lightweight human activity recognition, ArXiv Prepr. ArXiv230317845 (2023).
[66] A. Reiss, D. Stricker, Introducing a new benchmarked dataset for activity monitoring, Proc. - Int. Symp. Wearable Comput. Iswc. (2012) 108–109, https://doi.org/10.1109/ISWC.2012.13.
[67] J.R. Kwapisz, G.M. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, ACM SIGKDD Explor. Newsl. 12 (2011) 74–82, https://doi.org/10.1145/1964897.1964918.
[68] M. Gil-Martín, R. San-Segundo, F. Fernández-Martínez, J. Ferreiros-López, Time analysis in human activity recognition, Neural Process. Lett. 53 (2021) 4507–4525, https://doi.org/10.1007/s11063-021-10611-w.
[69] C. Han, L. Zhang, Y. Tang, W. Huang, F. Min, J. He, Human activity recognition using wearable sensors by heterogeneous convolutional neural networks, Expert Syst. Appl. 198 (2022), https://doi.org/10.1016/j.eswa.2022.116764.
[70] S. Xiao, S. Wang, Z. Huang, Y. Wang, H. Jiang, Two-stream transformer network for sensor-based human activity recognition, Neurocomputing 512 (2022) 253–268, https://doi.org/10.1016/j.neucom.2022.09.099.