Neural Processing Letters
https://doi.org/10.1007/s11063-022-10799-5

Deep Temporal Conv-LSTM for Activity Recognition

Mohd Halim Mohd Noor[1] · **Sen Yan Tan**[1] · **Mohd Nadhir Ab Wahab**[1]

Accepted: 7 March 2022

Abstract
Human activity recognition has gained interest from the research community due to the advancements in sensor technology and the improved machine learning algorithm. Wearable sensors have become more ubiquitous, and most of the wearable sensor data contain rich temporal structural information that describes the distinct underlying patterns and relationships of various activity types. The nature of those activities is typically sequential, with each subsequent activity window being the result of the preceding activity window. However, the state-of-the-art methods usually model the temporal characteristic of the sensor data and ignore the relationship of the sliding window. This research proposes a novel deep temporal Conv-LSTM architecture to enhance activity recognition performance by utilizing both temporal characteristics from sensor data and the relationship of sliding windows. The proposed architecture is evaluated based on the dataset consisting of transition activities—Smartphone-Based Recognition of Human Activities and Postural Transitions dataset. The proposed hybrid architecture with parallel features learning pipelines has demonstrated the ability to model the temporal relationship of the activity windows where the transition of activities is captured accurately. Besides that, the size of sliding windows is studied, and it has shown that the selection of window size is affecting the accuracy of the activity recognition. The proposed deep temporal Conv-LSTM architecture can achieve an accuracy score of 0.916, which outperformed the state-of-the-art accuracy.

Keywords Activity recognition · Deep learning · LSTM · Temporal model

1 Introduction

The rapid development of machine learning techniques and ubiquitous computing has spurred the interest from academia to analyze and interpret sensor data to extract knowledge from the omnipresent sensor over the previous few decades. The growing research community is interested in human activity recognition (later referred to as activity recognition) because of its tremendous usefulness in health monitoring, medical assistance, entertainment, and personal health tracking services. For instance, the real-time feedback from the activity detection system allows the healthcare professional to quickly monitor patients who require close

✉ Mohd Halim Mohd Noor
halimnoor@usm.my

[1] School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia

monitoring, especially those with body motion-associated diseases. Most of the research seeks to improve and boost the algorithm's accuracy, efficiency, and execution time by applying pattern recognition on the raw sensor data to extract valuable information related to the current activity for the user.

Activity recognition is one of several disciplines that utilize machine learning approaches to detect hidden patterns in sensor data to classify human activities. Thanks to the advancements in technology such as computer systems and power accessible today, the approaches have steadily improved. At the same time, the improved learning algorithm approaches also pave the way for the enhancement of activity recognition research. Traditional machine learning approaches, such as the Support Vector Machine (SVM) or Hidden Markov Model (HMM), have been widely utilized in many activity recognition-based studies. The underlying characteristics of the dataset must be manually retrieved using various feature extraction methods such as principal component and linear discriminant analysis [1], wavelet transform [2], homomorphic analysis [3] and local binary pattern [4], and fed into the machine learning algorithms for the learning algorithms to understand the patterns in the data. The disadvantage of the traditional machine learning approach is that researchers must demonstrate possession of the vast knowledge of the domain, which implies that the researchers must have a thorough grasp of the behavior and characteristics of the time-series data for better feature extraction. However, the process of feature extraction is still very susceptible to human mistakes.

Today, the learning algorithm has progressed from manual feature extraction to fully automatic feature learning by utilizing deep learning methods. Deep learning can extract features directly from data without human intervention. In recent years, many researchers have demonstrated and proved that the deep learning method is satisfactory [5]. One of the most important aspects of successful deep learning models is the network architecture. As deep learning methods like convolutional neural network (CNN) and recurrent neural network (RNN) become more sophisticated and refined in activity recognition, several researchers have advocated leveraging both methods. CNN is better at recognizing long-term repetitive activities, while RNN-based networks such as long short-term memory are better at recognizing short, natural-ordered activities [6]. By combining both main and mature deep learning methods, one may leverage the strengths of both methods to improve activity recognition performance.

In activity recognition, the activity signal is typically divided into segmentations or known as windows of equal size for subsequent feature extraction and classification. Typically, the window size is set based on hardware limitations and experience. Small window size would slice the activity signal into multiple separate segmentations. Thus, the segmentation lacks the information for activity recognition. On the other hand, large segmentation could contain multiple activity signals, confusing the classification model. In both cases, the segmentations do not have the optimal information of the activity signals, which would lead to misclassification. Another important property of the window segmentation is that they are inherently sequential due to the nature of human activities, whereby an activity window can be followed by a particular set of activity windows. For example, a window classified as standing is followed by either a standing window or a walking window only. However, the sequence of activity windows is often ignored in the development of classification models. The developed classification models consider only the current window which is the segmentation to be classified. Such models do not leverage the fact that the sequence of activity windows is inherently sequential due to the nature of human activities. Therefore, this work aims to develop a hybrid deep learning model that combines the strength of CNN and RNN to extract the salient feature representation and capture the temporal information in the activity data.

Unlike previous hybrid models where the input is a single-window segmentation, the proposed model accepts a sequence of activity windows to model the dependencies between the windows. Each activity window is processed by a separate stream that extracts the local window features. The window features are then concatenated to become a sequence of window features. Then, the dependencies of the features are modeled to capture a better representation of the data to improve the model generalization.

The remainder of this paper is organized as follows. Section 2 reviews the related works. In Sect. 3, we present the proposed methodology that consists of data collection and preprocessing, the proposed hybrid model and the implementation details. Section 4 presents the experimental results and their discussion. Finally, the conclusions are presented in Sect. 5.

2 Related Works

Deep learning method has been widely implemented to overcome the limitation of machine learning. Deep learning can extract features automatically, which leads to lesser human effort. Numerous deep learning models have been proposed for activity recognition, including CNN models, RNN models and hybrid CNN and LSTM models.

2.1 CNN Models

In [7], a CNN model is designed to take in raw accelerometer data in three-dimensional (3D) directly without any complex pre-treatment. Before feeding to the first convolution layer, the input is pre-processed with the sliding window method before normalization is applied to the accelerometer data. The normalized data is then fed into 1D convolution and max-polling layer. The author proposed to perform validation on the model based on the benchmark WISDM dataset. The experimental result indicates that the proposed model can achieve high accuracy while maintaining low computation costs. A multiple channel CNN was presented as a solution to the problem of activity recognition in the context of exercise programmes [8]. A self-collected dataset comprised of 16 activities from the Otago exercise program is captured and used in this experiment. Multiple sensors are placed across body parts to capture the raw inertia data for various activities which each sensor will be fed into a separate CNN channel. The results from all sensors after CNN's operation will be compared individually to determine the best location to place sensors for better lower-limb activity detection. In this experiment, the authors also conclude that multiple sensor combinations can produce better results than a single sensor source.

A Deep Human Activity Recognition model is proposed, which converts the motion sensor data into a spectral image sequence before feeding these images into two independently trained CNN models [9]. Each CNN model takes in the image sequences that are generated from the accelerometer and gyroscope. The outputs of the trained CNNs are then fused to predict the final class of human activity. In this experiment, the public dataset Real-world Human Activity Recognition (RWHAR) is used. This dataset contains eight activities which are climbing stairs down and up, lying, standing, sitting, running/jogging, jumping and walking. The proposed model can achieve an overall F-score of 0.78 for both static and dynamic activities and 0.87 for dynamics activities. The author also claimed that this model is capable of handling image input directly. The model's generalization is encouraging; however, the recognition accuracy is not comparable with the other benchmark deep learning model. In [10], three strategies are proposed to exploit the temporal information of a sequence

of windows. The first strategy is to compute the average of the windows which will be used as input to the CNN model. In the second strategy, the sequence of windows is fed to a concurrent CNN, and the activity class is determined based on the average scores. The final strategy is similar to the second strategy. However, the learned features are combined using global average pooling layer to produce the final prediction.

Instead of a single classifier of CNN, an ensemble of CNN has been proposed to improve the accuracy of activity recognition. Zhu et al. [11] proposed a human activity recognition framework based on CNN using a fusion of various smartphone-based sensors such as accelerometer, gyroscope, and magnetometer. The proposed framework is an ensemble of two different CNN models whereby the first CNN is trained to predict the activity classes while the second CNN is trained to focus on the activity classes that have a high number of misclassification. The output of individual CNN models is then combined using weighted voting to predict unknown activities. The experimental result indicates that this proposed model can achieve up to 0.962 in terms of accuracy. Zehra et al. [12] also proposed an ensemble model consisting of three different CNN models. The ensemble model averages the three CNN models' outputs to produce the final prediction. The authors evaluated the performance of each CNN model before ensembling each CNN model for overall performance validation. The experimental result indicates that the performance of the ensemble model is better than the three CNN models. The ensemble model achieved an accuracy of 0.940. This experiment shows that the ensemble learning model can generalize how the learning effect of the weak learner could be boosted and improve the overall model. In [13], a two-channel CNN model is proposed for activity recognition. The proposed model leverages the frequency and power characteristics extracted from sensor signals to improve recognition accuracy. The model was validated on a public UCI-HAR dataset and demonstrated an accuracy of 0.953. The downside of this method is that it requires the extraction of specific features to improve activity recognition from sensor data. The performance of CNN model is enhanced by integrating the attention mechanism module to determine the relevance of the features [14]. To extract the local features, the three acceleration channels are fed to three concurrent convolutional layers with different filter sizes. Then the attention mechanism computes the contribution of the features to select the relevant features. The model was validated on a public WISDM dataset and demonstrated an accuracy of 0.964.

Based on the aforementioned studies, it can be observed that most implementations did very well at classifying the activities. They could automatically extract the salient features, which leads to good classification performance. However, the temporal information of the sensor data is not leveraged for activity classification.

2.2 RNN-Based Models

Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two popular RNN variations. Several studies employ the RNN architectures to tackle the activity recognition problem. Chen et al. [15] presented a feature extraction approach based on LSTMs for activity recognition. In the study, the accelerometer data is segmented into a sequence of windows of size N, and the three acceleration channels are individually processed. Thus, three LSTMs are used to perform feature extraction on the windows. Following the LSTMs, a concatenation operation is performed to produce a feature vector which will be fed to a softmax classifier. WISDM dataset is used to validate the proposed model. The experimental results show that the proposed model achieved an accuracy of 0.921. A similar work is reported in [16], whereby two layers of LSTMs are proposed to perform feature extraction

on accelerometer and gyroscope data. The results show that the proposed model achieved an average accuracy of 0.920. Furthermore, it has been shown that batch normalization can attain the same accuracy with nearly four times fewer training epochs.

Other than employing the LSTM to process the sensor data directly, several models of ensemble LSTM networks have been proposed to improvise the accuracy of activity recognition. The performance of ensembles of deep LSTM is verified and reported in [17]. The authors built diverse base learners using LSTM and the predictions of the base learners are combined via average operation to obtain a more robust and improvised classification performance. The authors also proposed a modified training procedure such as random sampling with varying lengths of the sensor data, and sample-wise model evaluation is performed during inference. The authors validated the proposed model on three different datasets: Opportunity, PAMAP2, and Skoda. The results show that the ensemble model achieved an accuracy of 0.726, 0.854 and 0.924 for Opportunity, PAMAP2, and Skoda respectively. Also, the experimental results indicate that the ensemble model performs better than a single classifier. Li et al. [18] proposed an ensemble model using LSTMs to accept input segmentation with different sizes to model the underlying temporal patterns at various degrees of granularity. The predictions of the LSTMs are combined via element-wise multiplication to produce the final prediction. The experimental results show that the proposed model achieved an average accuracy of 0.961.

Mahmud et al. [19] proposed a multi-stage LSTM-based model to process multimodal sensor data for activity recognition. This proposed model consists of three key components, which are temporal feature extractor, temporal feature aggregator and global feature optimizer. The temporal feature extractor comprises two layers of LSTM to extract temporal features from each sensor data. The temporal feature aggregator aggregates the temporal features, taking into account both the time-axis and the feature-axis to preserve the temporal relationship. The global feature optimizer consists of three layers of LSTM to extract global features from the aggregated temporal features. The experimental results show that incorporating multiple sensors into the proposed model outperformed the single sensor-based model.

Although RNN networks have been shown capable of modeling the temporal characteristic of sensor data, it is generally not performing well in extracting local features from sensor data. Therefore, there is a need to combine CNN with LSTM to exploit the strength of both deep learning methods.

2.3 Hybrid Models

In recent years, hybridization of CNN and RNN networks has been experimented with to improve the performance of activity recognition. Various hybrid deep learning models have been proposed in previous studies. But the focus of this study is the hybrid models of CNN and RNN. Ordóñez & Roggen first proposed a novel DNN framework for activity recognition, consisting of four convolutional layers, followed by two recurrent layers and a softmax layer as a classifier [20]. The convolutional layers are used as a feature extractor to produce the feature representation of the sensor data. In contrast, the recurrent layers are used for modeling the temporal dynamics of the feature maps. This proposed framework employs the sliding window approach to segment the time series data. The proposed model is validated on two popular public datasets, which are OPPORTUNITY and SKODA. The accuracy for OPPORTUNITY and SKODA are 0.930 (for modes of locomotion with no null class) and 0.958 respectively.

Mekruksavanich and Jitpattanakul [21] proposed a similar hybrid CNN-LSTM model. In the study, the authors added Bayesian Optimization to fine-tune each LSTM and CNN network parameter. The author evaluated the proposed model using WISDM public data. The result of the experiments indicates that the proposed model outperformed the other baseline, achieving an average accuracy and F-score of 0.962 and 0.963 respectively. A similar CNN-LSTM model for activity recognition is reported in [22]. However, the authors proposed to wrap the convolutional and pooling layers with a Time Distributed wrapper to maintain its temporal integrity for the LSTM layers. The input data is reshaped to 3D as required by the Time Distributed wrapper. The proposed model achieved an accuracy of 0.921 and 0.991 for iSPL and UCI-HAR datasets respectively. Thus, it is concluded that the proposed model outperformed other deep learning models that simply use the raw sensor data as input. Another similar model is reported in [23]. The input dimension is first expanded to obtain heterogeneous data and the data is then fed to the proposed model. The proposed model achieved an accuracy of 97.65% on the UCI-HAR dataset.

Wang et al. [24] proposed a similar CNN-LSTM architecture in which the focus is to model the transition of the activities of the window sequence. To achieve this, the author proposed to treat the sensor data as an image-alike 2D array. The image alike array data is fed into a three-layer CNN network for automatic feature extraction to obtain the feature vector. The feature vector from the previous layer is then fed into LSTM layers to model the relationship between time and action sequence. The proposed model is validated using the SBHAPT dataset and the experimental results show that the proposed achieved an accuracy of 0.959. The limitation of this proposed model is the pre-requisite of treating the signal data as image-like, which incurs an additional pre-processing step to convert the raw real-time signal into image-like form before feeding into the proposed model.

Singh et al. [25] proposed a deep neural network architecture that consists of CNN, LSTM and a self-attention mechanism. The CNN and LSTM layers extract spatio-temporal features from multiple time-series data, and the self-attention layer is utilized for training on the most significant time point. The proposed model is validated with different data sampling strategies on six public datasets, which are mobile health (MHEALTH), USC human activity dataset (USC-HAD), Wireless Sensor Data Mining (WISDM), UTD Multimodal Human Action Dataset (UTD-MHAD2), Wearable Human Activity Recognition Folder (WHARF), and UTD Multimodal Human Action Dataset (UTD-MHAD1). The proposed model achieved an accuracy of 0.949, 0.909, 0.904, 0.898, 0.824 and 0.580 for the six datasets above respectively. The proposed model also indicates that the self-attention mechanism significantly improvises the performance of the model. The results show that the proposed architecture has significantly outperformed the state-of-the-art methods. However, the experiments did not involve transitional activities such as stand-to-sit, sit-to-stand and sit-to-lie.

Abdel-Basset et al. [26] presented a supervised dual-channel model comprised of LSTM and an attention mechanism. The long-term temporal representations of the sensor data are modeled by the LSTM. An advanced residual network, on the other hand, effectively extracts hidden characteristics from high-dimension sensory input. The attention mechanism is applied on LSTM to further improvise on the temporal fusion performance. The proposed model for multichannel spatial fusion also includes a novel adaptive-squeezing CNN. The proposed model is evaluated on two benchmark datasets: UCI-HAR and WISDM. The results show that the proposed model outperforms existing state-of-the-art models by achieving an accuracy of 0.977 and 0.989 for UCI-HAR and WISDM respectively.

Xia et al. [27] proposed a hybrid deep learning architecture that is made up of two layers of LSTM followed by convolutional layers. The global average pooling layer (GAP) is applied instead of the fully connected layer after the convolutional layers and followed with

a batch normalization layer (BN). The authors found that GAP could help to reduce the model parameters while the batch normalization layer helps to speed up the convergence. The proposed model is validated on three different datasets: OPPORTUNITY, WISDM and UCI-HAR, achieving an accuracy of 0.927, 0.958 and 0.958.

Nafea et al. [28] proposed a novel hybrid model to extract the temporal cues from the sensor data. The proposed model consists of a two-stream of CNN and BiLSTM. The features from both streams are then concatenated and fed to a fully-connected layer for classification. The proposed model is evaluated on WISDM and UCI-HAR datasets and the results show that the proposed model outperformed the state-of-the-art models, achieving an accuracy of 0.985 and 0.971 respectively. The authors claim that CNN-BiLSTM is an efficient solution to extract spatial and temporal features. Similar work is reported in [29], whereby a novel hybrid model is proposed to extract local features and global temporal relationship of the features. The proposed model consists of a two-stream of convolutional layers and LSTM-based attention mechanism modules. The proposed model is evaluated on WISDM, UCI-HAR, Opportunity and PAMAP2 datasets. The results show that the proposed model outperformed the state-of-the-art models, achieving an average accuracy of 0.975. Shi et al. proposed a similar model for WiFi-based activity recognition. The proposed model consists of a series of convolutional and max-pooling layers followed by a bidirectional LSTM and an attention mechanism module. The activities considered in the experiments are standing, sitting, walking, running, stand up and sit down. The results show the proposed model significantly improves the recognition accuracy.

Gao et al. [30] proposed a novel hybrid model to capture both the channel-wise and temporal dependencies of the sensor data. The segmented sensor data is fed to convolutional layers to extract the feature representation. The features is then fed to a squeeze-and-excitation module which consists of channel attention submodule and temporal attention submodule to model the dependencies. The channel attention submodule consists of a two-stream of max-pooling layer and average pooling layer, and each pooling layer is followed by a fully-connected layer with ReLU activation function. The outputs of the fully-connected layers are then concatenated via the temporal axis and converted to probabilities using Sigmoid function. The temporal attention submodule has similar two-stream network. But the concatenation is performed along the channel index. The proposed model is evaluated on four different datasets: WISDM, UniMiB SHAR, PAMAP2 and Opportunity. The experimental results show the proposed model achieved better performance than the existing models.

Based on the past literature, hybrid models can achieve a satisfactory result. However, several limitations are posed by the aforementioned studies. First, the studies do not exploit the temporal information of the sequence of activity windows. Also, most of the studies except [24] consider only basic activities such as walking, standing and sitting and ignore the transitional activities such as stand-to-sit, sit-to-stand which have a much shorter duration and less occurrence. Therefore, this paper presents a hybrid deep learning model consisting of three parts: feature learning pipelines, sequential learning module and activity classifier. The feature learning pipeline consists of a concurrent feature extraction module that accepts a sequence of activity windows to learn feature representation of the windows, while the sequential learning module model the temporal dependencies between the windows. The temporal features produced by the sequential learning module are fed to the classifier for activity recognition.

3 Proposed Methodology

3.1 Data Collection and Pre-processing

The dataset that is used in this study is the Smartphone-Based Recognition of Human Activities and Postural Transitions (SBHAPT) dataset [31]. The dataset is publicly available from UCI and is widely used by numerous researchers to validate their architecture. The rationale of the dataset selection is due to its data collection method whereby the subjects performed the activities continuously. Thus, the dataset contains not only the basic activities, but also the transitions between two activities. As the aim of this study is to exploit the sequence of activity windows, this characteristic becomes a critical component to evaluate our proposed model. To the best of our knowledge, this is the only public dataset that contains basic activities as well as their transitions. The dataset was collected from 30 subjects and each subject performed the protocol twice. During the data collection, a smartphone integrated with a tri-axial accelerometer and tri-axial gyroscope is attached to the waist of the subjects. The sensor data is generated at a constant rate of 50 Hz.

A total of 12 activities are captured in this dataset, including the basic activities and the postural transitional activities. Among the six basic activities, three of them are static activities such as standing, sitting, lying and the other three are dynamic activities such as walking downstairs, walking upstairs and walking. The transitional activities are sit-to-lie, lie-to-sit, stand-to-sit, stand-to-lie, lie-to-stand, and sit-to-stand. Note that stand-to-lie and lie-to-stand consist of two transitional activities. For example, the stand-to-lie is composed of stand-to-sit followed by sit-to-lie. However, the original authors of the dataset annotate the activities as a single transitional activity. Table 1 lists the activities and their number of samples. The sensor data is normalized to zero mean and unit variance. Then the sensor data is segmented using the fixed-size sliding window method. An activity window may contain samples from two activity classes. This is due to the nature of time-series data as the subject transition from one activity to another. Therefore, the activity windows are labeled according to the majority samples within the window. For example, if the size of the window is 100

Table 1 Distribution of sensor data (number of activity samples)

ID	Activity	Number instances	Percentage (%)
A1	Walk	122,091	14.97
A2	Upstairs	116,707	14.31
A3	Downstairs	107,961	13.24
A4	Sit down	126,677	15.53
A5	Stand	138,105	16.93
A6	Lie	136,865	16.78
A7	Stand to sit	10,316	1.26
A8	Sit to stand	8029	0.98
A9	Sit to lie	12,428	1.52
A10	Lie to sit	11,150	1.37
A11	Stand to lie	14,418	1.77
A12	Lie to stop	10,867	1.33
Total		815,614	100.00

samples, and 55 samples belong to class A and the remaining samples belong to class B. The window is labeled as class A.

3.2 Proposed Deep Temporal Model

The block diagram of the proposed model is as illustrated in Fig. 1. The time-series data generated by the sensor is segmented with the sliding window method. Each activity window contains sensor data that lasted for a finite amount of time. The proposed model accepts a sequence of activity windows as input. The window sequence contains K previous activity windows in addition to the current window to be predicted. The K previous windows provide additional information to the model in predicting the current window. It is worth noting that in the figure, the sensor data is segmented with the sliding window with no overlapping. However, overlapping segmentation is typically used and has been shown to achieve better recognition accuracy as reported in previous studies [32]. Furthermore, the overlapping sliding window increases the number of segmentations, improving the generalization of the deep learning models.

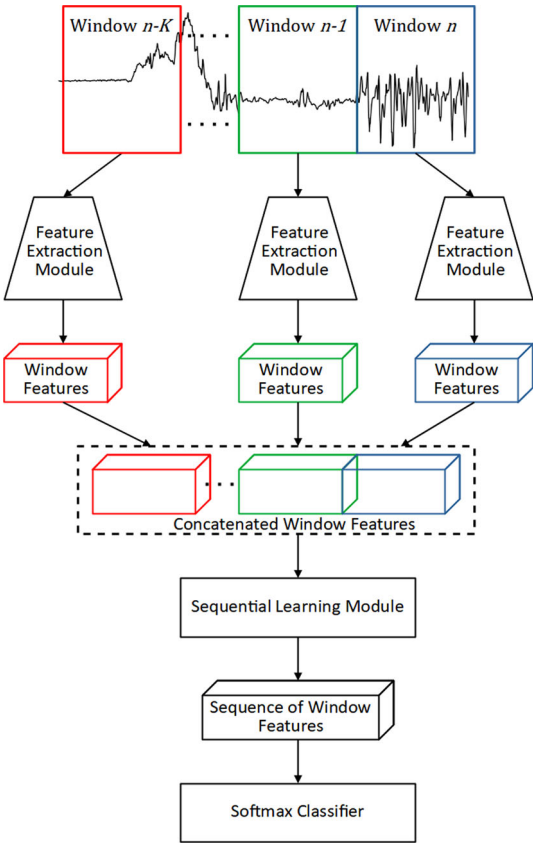

Fig. 1 The block diagram of the proposed hybrid model

Table 2 The parameters of the feature learning pipeline (segmentation size equals 128)

Layer	Kernel or pool size	Stride	Activation	Output shape
Input				128 × 6
1D Conv	2	1	tanh	128 × 8
Max pooling	2	1		127 × 8
Dropout (prob = 0.5)				
1D Conv	2	1	tanh	127 × 8
Max pooling	2	1		126 × 18
Dropout (prob = 0.5)				
1D Conv	2	1	tanh	126 × 36
Max pooling	2	1		125 × 36
Dropout (prob = 0.5)				

The first part of the proposed model is the concurrent feature learning module that acquires the sequence of the activity windows. Each feature learning pipeline is composed of convolution and pooling operations with dropout regularization. The convolution layers are used to extract low- and high-level features in a hierarchical manner. The hyperbolic tangent (tanh) activation function is selected based on the experiments that have been conducted. The pooling layer reduces the size of the feature maps after each convolution layer which yields a reduction in the computational complexity. The maximum pooling with pool size equals 2 and stride equals 1 is used because it has been shown to be effective for sensor-based activity recognition [24]. The dropout regularization is applied after the maximum pooling layer to reduce overfitting and improve the model generalization. Table 2 lists the parameters of the feature learning pipeline.

As can be shown in Fig. 1, each feature learning pipeline concurrently processes different activity windows and produces the local window features. The window features are the feature representation of the activity windows that are segmented at different times. These window features are concatenated to form a single window feature before being used as input to the sequential learning module. The concatenation of the window features represents the feature representation of the activity windows segmented at different times. Thus, the sequential learning module models the dependencies of the activity windows. Assuming the feature maps of the feature extractor is denoted by $\boldsymbol{x}_n$ where n denotes n th window or the window to be predicted. The concatenation of the window features or window sequence is given as follows:

$$\boldsymbol{z} = \left[\boldsymbol{x}_{n-K}, \ldots, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n\right] \tag{1}$$

where K is the number of previous windows being used for predicting the window n. The size of the window sequence vector is given as follows:

$$T = L \times (K + 1) \tag{2}$$

where L is the size of the single-window feature.

The window sequence is then fed to the sequential learning module. The sequential learning module aims to model the dependencies between the window features. Previous works have shown that LSTM is effective in modeling time series data for activity recognition. Therefore, the LSTM network is adopted as the sequential learning module. LSTM networks

have the form of a chain of repeating neural network-based modules known as LSTM cell. Each cell accepts a single feature as input; thus, the number of LSTM cells is determined by the size of the window sequence vector. This input is processed, and an output is produced which will be fed to the next cell. In this way, the temporal information in the window sequence is captured for classification.

An LSTM cell consists of several gating units that control the flow of information from one LSTM cell to another LSTM cell. The first gate is called the 'forgetting gate'. This gate examines the input feature and the previous cell's output, and determines which information needs to be filtered out from the cell. This operation is performed by a layer with sigmoid activation function, which outputs the value in the range of 0 (filter out) and 1 (keep).

$$f_t = \sigma\left(w_f \cdot \left[h^{t-1}, z^t\right] + b_f\right) \tag{3}$$

The second gate is called the 'input gate'. This gate is responsible for storing information in the cell based on the input feature and the output of the previous cell. This operation is performed by two layers, one with sigmoid activation function and the other layer with tanh activation function. The sigmoid layer retrieves the relevant information to be used to update while the tanh layer creates a candidate vector that will be used to update the cell state.

$$u_t = \sigma\left(w_u \cdot \left[h^{t-1}, z^t\right] + b_u\right) \tag{4}$$

$$\tilde{C}_t = \tanh\left(w_c \cdot \left[h^{t-1}, z^t\right] + b_c\right) \tag{5}$$

The computation to update the cell state is given as follows:

$$C_t = u_t * \tilde{C}_t + f_t * C_{t-1} \tag{6}$$

As can be seen from the above formula, the update considers the state of the previous cell. This allows the cell to add some relevant information from the previous cell state to the cell state.

The final gate is the 'output gate'. This gate examines the input feature and the previous cell's output and produces the output that is based on the cell state. The computation of the output is given as follows.

$$v_t = \sigma\left(w_v \cdot \left[h^{t-1}, z^t\right] + b_v\right) \tag{7}$$

$$h_t = v_t * \tanh(C_t) \tag{8}$$

Each of the LSTM cells is set to have 48 hidden units and the LSTM network returns only the output of the last cell, h_T. The dropout regularization with a dropout rate equal to 0.5 is applied to the LSTM network to improve the model generalization. Finally, the output of the LSTM network is fed to a softmax classifier with 12 units whereby each unit represents an activity class.

Given a sequence of windows, $\boldsymbol{x}_{n-K}, \ldots, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n$, the model training is performed by minimizing the loss function, L between the prediction and the window's label. This can be expressed as follows.

$$(\boldsymbol{w}, \boldsymbol{b}) = \underset{\boldsymbol{w}, \boldsymbol{b}}{\arg\min}\, L\left(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n\right) \tag{9}$$

where $\boldsymbol{y}_n$ is the label of the current window (window being predicted) and $\hat{\boldsymbol{y}}_n$ is the prediction of the current window. The loss function is the cross entropy, which is defined as follows.

$$L\left(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n\right) = -\sum_{m=1}^{M} \boldsymbol{y}_n \log \hat{\boldsymbol{y}}_n \tag{10}$$

3.3 Implementation Details

The dataset is split using the subject-based hold-out method. The split ratio is 22 subjects to 8 subjects. Note that each subject was asked to perform the protocol twice. Therefore, there are 44 activity data for the training set and 16 activity data for the test set. In the experiments, the validation set is not used due to the limitation of the dataset. Therefore, the whole training set is used to train the model, and the test is used to evaluate the model. The training epoch is set to 500 and the batch size is set to 128. The training loss is monitored during training, and the model checkpoint is used to save the best weights. The proposed model is trained to minimize the cross-entropy loss. The training algorithm is the adaptive moment estimation (Adam) optimizer. The L2 regularization is used to prevent the model from overfitting the training data. The proposed model was implemented using the TensorFlow framework. The workstation is equipped with Intel i5, 16 Gb memory and Nvidia GTX 1070. Several performance metrics are used to evaluate the performance of the proposed model. The performance metrics are precision, recall, F-score and accuracy. The precision indicates the ability of the model to distinguish an activity class from all the other classes. The recall indicates the ability of the model to correctly recognize an activity class. The F-score is the average of recall and precision. The accuracy indicates the fraction of correctly classified activity windows.

4 Experimental Results

4.1 Experimental Setup

This section describes the experimental results of this study. Two experiments have been conducted to evaluate the performance of the proposed model. First, we experimented with the relation between the number of feature learning pipelines and the recognition accuracy. In this experiment, we set the size of the window segmentation to 120 samples with an overlapping of 60 samples. This results in 9237 and 4354 windows for the training set and test set respectively. The number of feature learning pipelines is increased from 1 to 4. The second experiment involved the effect of window size on recognition accuracy. Based on a study reported in [33], the optimal window size for recognizing energetic and non-energetic activities is in the range of 1–5.75 s, while the recommended window size to prioritize recognition speed is in the range of 0.25–3.25 s. Therefore, in this experiment, we experimented with five window sizes in the range of 80 samples (1.6 s) and 140 samples (2.8 s) as shown in Table 3.

The recall, precision and accuracy are used to determine the optimal parameters and evaluate the proposed model. Recall is defined as the ability of the model to identify the activity class of a window segmentation. Precision is the ability of the model to distinguish an activity class from all the other classes. Accuracy is the fraction of correctly classified

Table 3 The number of window segmentations for each experimental setup

Number of windows and labels	80 samples	100 samples	120 samples	140 samples	128 samples
Training set	13,856	11,084	9237	8659	7917
Test set	6532	5225	4354	4082	3732

window segmentation. The performance metrics are given as follows.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

4.2 Number of Feature Learning Pipelines

In this experiment, the optimal number of feature learning pipelines is determined. Five models are built as listed in Table 4. The models are trained using the training set until the number of maximum epochs is reached. During the training, the training accuracy is monitored, and the best weights are saved based on the training accuracy. The trained models are then evaluated using the test set. First, the proposed model is evaluated with 1 feature learning pipeline without the sequential learning module. This is to evaluate the ability of the feature extractor in extracting the relevant features for activity classification. The results show that the proposed model is able to achieve high accuracy of 0.900. Then, we evaluate the performance of the model when the sequential learning module is integrated to learn the temporal information of the sensor data as well as the sequence of the activity windows. An improvement of 0.003 is observed. This shows that capturing the temporal information of the sensor data and activity windows is significant in the classification of the activities. Following this, the number of feature learning pipelines is increased to two to model two activity windows in sequence. In other words, the proposed model utilizes the previous activity window in predicting the current window. Note that, for models with multiple feature learning pipelines, the number of windows is equal to the number of feature

Table 4 The accuracy of the proposed model with the different number of feature learning pipelines

Model	Accuracy
1-feature learning pipeline	0.900
1-feature learning pipeline with the sequential learning module	0.903
2-feature learning pipeline with the sequential learning module	0.905
3-feature learning pipeline with the sequential learning module	0.912
4-feature learning pipeline with the sequential learning module	0.904

learning pipelines. However, the number of output (prediction) remains the same, which is the prediction of the current window (window being predicted). This is indicated by formula (9). The results show that the performance of the proposed model is increased by 0.002. The experiment proceeds with three and four feature learning pipelines. The best performance is observed when the proposed model is integrated with three feature learning pipelines with an accuracy of 0.912. For the next experiment, this model configuration is used to investigate the optimal window size for activity recognition. Table 4 list the recognition accuracy for each model configuration.

4.3 Window Size

In this experiment, the proposed model with 3-feature learning pipeline is used to determine the optimal window size. The models are trained using the training set until the number of maximum epochs is reached. During the training, the training accuracy is monitored, and the best weights are saved based on the training accuracy. The trained models are then evaluated using the test set. The window size is varied to 80, 100, 120, 140 and 128 samples. Each window size has 50% overlapping. This experiment is critical in determining the best window size due to the characteristic of the sensor data, which directly affects the activity classification. Some activities take a longer time to complete, while others are completed in a short time. A too small or large window size may cause the window to be wrongly classified. This problem would be compounded when a sequence of activity windows is considered during the activity classification. Figure 2 shows the recall, precision and F-score measures of the activity recognition with the different windows sizes. Table 5 lists the accuracy of the experimental setups.

Overall, it is observed that the proposed model performed well in classifying the non-transitional activities (A1–A6) compared to transitional activities (A7–A12). All the non-transitional activities were classified with F-score measures above 0.800, whereas the transitional activities were classified with F-score measures in the range of 0.403 and 0.753. This is due to the limited windows of transitional activity. This is shown in Table 1 whereby the number of samples of non-transitional activities is significantly higher than the number of samples of transitional activities. It is also observed that a significant number of A4 windows are misclassified as A5 and vice versa in each of the experiments. Please refer to the "Appendix" for the confusion matrix. This is due to the fact that both activities have similar signal patterns. Hence, the features that have been learned might have similar representations.

Experimental setup 1 (window size equals 80 samples) shows that the proposed model performed well in classifying the non-transitional activities compared to transitional activities with activity A3 achieved the highest precision of 1.00. The recognition accuracy of the experiment is 0.892. Experimental setup 2 (window size equals 100 samples) shows similar performance in classifying the non-transitional activities compared to the transitional activities. However, it is observed that there is a slight improvement in classifying activity A1 to A6. It is also observed that the number of misclassifications of activity A4 and A5 is slightly lower. Overall, the recognition accuracy of the proposed model is 0.897. Experimental setup 3 (window size equals 120 samples) shows better performance in terms of F-score measures in all activities except A3, A8, A9 and A12. The classification of activity A4 and A5 is also improved. The accuracy of the proposed model is 0.907. Experimental setup 4 (window size equals 140 samples) shows a significant reduction in the recognition accuracy whereby the accuracy is 0.891, which is 0.016 lower than the experimental setup 3. Similar

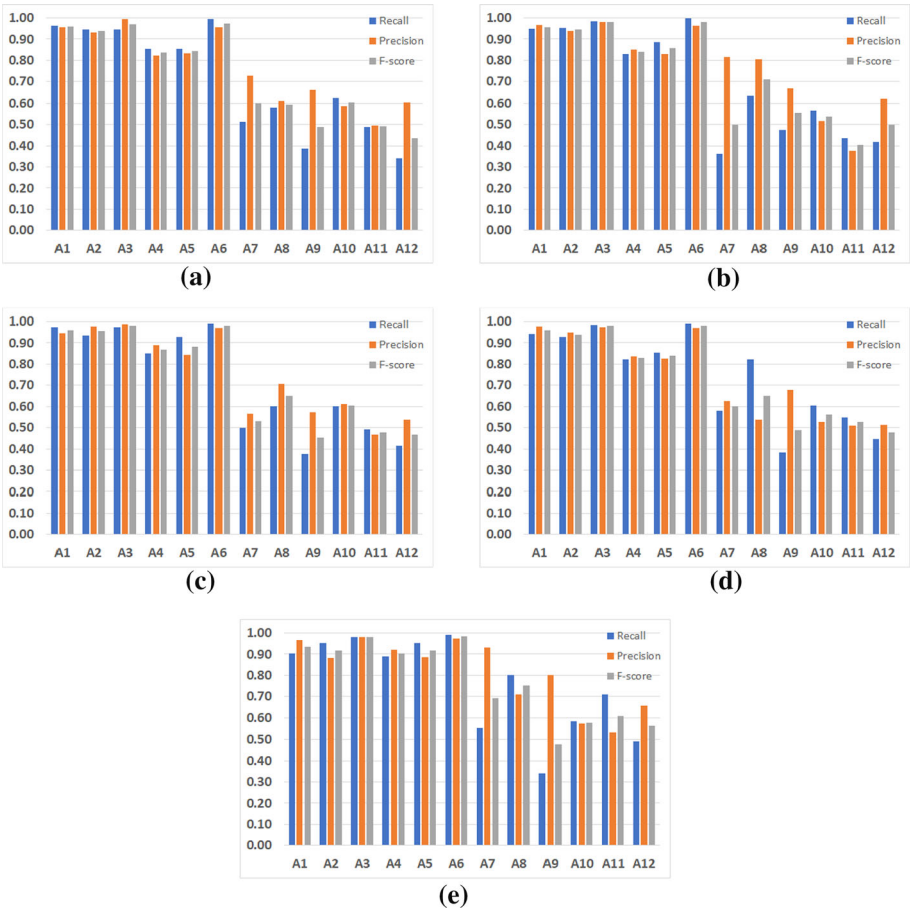

Fig. 2 The performance measures of activity recognition with different window sizes **a** 80 samples, **b** 100 samples, **c** 120 samples, **d** 140 samples and **e** 128 samples

Table 5 Accuracy of activity recognition with different window sizes

	80 samples	100 samples	120 samples	140 samples	128 samples
Accuracy	0.892	0.897	0.907	0.891	0.916

classification pattern is observed whereby the non-transitional activities achieved better F-scores compared to the transitional activities. Based on experimental setup 4, it is concluded that the optimal window size is between 120 and 140. Therefore, another experiment was carried out by setting the window size equal to 128 samples. Experimental setup 5 shows a significant improvement in accuracy which is recorded at 0.916, which is 0.008 higher than experiment setup 3. It is observed that the number of misclassifications of activity A4 and A5 is decreased. Also, the transitional activities (activity A7–A12) achieved better F-score measures except for activity A10. Therefore, we can conclude that a window size of 128

is the optimal value for activity recognition. Next, we compare the proposed model with a benchmark hybrid model.

4.4 Comparison with the State-of-the-art Models

We compare the proposed model with other state-of-the-art models. Table 6 reports the summary of the state-of-the-art models. For fair comparison, the table shows the model performance in classifying datasets of basic activities only such as jogging, walking, walking upstairs and downstairs, standing, sitting, lying down. The performance of the state-of-the-art models on various datasets is shown in the table. The performance of our proposed model is comparable, if not better than the state-of-the-art models.

The state-of-the-art models were evaluated on datasets collected from subjects while performing the activities separately (not in continuous manners). Thus, the datasets contain only basic activities but no transitional activities. Also, the studies split the datasets by instance (sample) except for study [24, 26]. Hence, the intra subject dependencies is present in the training set which would inflate the recognition accuracy. Unlike the state-of-the-art models, our proposed model is evaluated on a dataset with basic activities and the transitions between the activities. Classifying the dataset is challenging because the window segmentation might contain data belonging to different activities. The dataset is split by subject.

As shown in the table, all the state-of-the-art models employed convolutional layers and LSTM to extract local and temporal features for more accurate recognition results. Various improvements to the classification model have been proposed to improve recognition accuracy. The attention mechanism module is integrated into the model to learn the relevance of the features for prediction [25, 26]. In [30], the squeeze-and-excitation-based module is integrated to model the dependencies of the feature maps. Although the modules are shown to improve the recognition accuracy slightly, the integration increases the complexity of the model. Furthermore, the state-of-the-art models do not exploit the sequence of the activity windows when performing the recognition since the models accept a single window segmentation as the input. Unlike the state-of-the-art models, our proposed model accepts a sequence of activity windows which allows the relationship of the window features to be modeled and consequently improves the recognition accuracy. In terms of the number of parameters, our proposed model has the least number of parameters compared to the state-of-the-art models.

In the experiment, a comparison of the proposed model with a benchmark model is performed. The benchmark model is the hybrid convolution LSTM model proposed by [24]. The rationale behind the comparison is that the authors used the same public SBHAPT dataset in their study. However, the method reported in [24] converted the sensor data into image form before feeding it into the proposed model. Therefore, to perform a fair comparison, we built and trained the benchmark hybrid model on the SBHAPT dataset. The parameters of the benchmark model such as kernel size, LSTM unit, training epoch, optimizer were set and defined according to the study. The benchmark model was trained and evaluated with the same training and test ratio. The performance measures of the benchmark model are given in Fig. 3a. The comparison of the performance metrics is given in Fig. 3b and Table 7.

We observed that our proposed model with three feature learning pipelines outperforms the benchmark architecture in terms of accuracy. The accuracy of the proposed model is 0.916 which is 0.013 higher than the benchmark model. In terms of recall, precision and F-score, the proposed model performed better in classifying the non-transitional activities (A1–A6), achieving an average F-score of 0.939 which is 0.014 higher than the benchmark model. However, the proposed model achieved a slightly lower average F-score measure in

Table 6 Summary of the state-of-the-art models

Relevant Model	Model performance	Description
Singh et al. [25]	Dataset preparation: Leave-one-out (subject-independent) Accuracy MHEALTH: 0.9486 USC-HAD: 0.9088 UTD-MHAD2: 0.8994 WISDM: 0.9041	The proposed model accepts a single window segmentation for activity recognition. The architecture of the proposed model consists of convolutional layers to extract local features, followed by LSTM layer to capture the temporal dependencies of the features and finally followed by an attention mechanism to assign different weights to the features to indicate the relevance of the features for classifying the activity Number of parameters: N.A
Abdel-Basset et al. [26]	Dataset preparation: N.A Accuracy UCI-HAR: 0.9770 WISDM: 0.9890	The proposed model accepts a single window segmentation for activity recognition. The architecture of the proposed model consists of a two-stream of spatial feature extractor and temporal feature extractor, whereby a series of residual blocks is used to extract the spatial features while LSTMs with attention mechanism are used to extract and assign weights to the temporal features Number of parameters: 312,934
Xia et al. [27]	Dataset preparation: Hold-out with 7:3 ratio (subject-independent) F-score UCI-HAR: 0.9578 WISDM: 0.9585	The proposed model accepts a single window segmentation for activity recognition. The architecture of the proposed model consists of two layers of LSTMs to extract the temporal features of the data, followed by convolutional and max-pooling layers to extract the local features Number of parameters: 49,606
Nafea et al. [28]	Dataset preparation: N.A Accuracy WISDM: 0.9853 UCI-HAR: 0.9705	The proposed model accepts a single window segmentation for activity recognition. The architecture of the proposed model consists of two-stream of convolutional layers and bi-directional LSTM to extract local features and temporal features, respectively. Finally, the features are concatenated for classification Number of parameters: N.A
Gao et al. [30]	Dataset preparation: Hold-out with 7:3 ratio Accuracy WISDM: 0.9885 UniMiB: 0.7903	The proposed model accepts a single window segmentation for activity recognition. The proposed model has two squeeze-and-excitation-based modules: temporal attention and channel-wise attention to capture the temporal and channel-wise dependencies of the features extracted by convolutional layers, respectively Number of parameters: 950,000–3,510,000
Proposed model	Dataset preparation: Hold-out with 7:3 ratio (subject independent) Accuracy SBHAPT: 0.9160	The proposed model accepts multiple window segmentations for activity recognition. The proposed model has concurrent feature learning pipelines which consist of convolutional and max-pooling layers to extract local window features. The window features are concatenated and modeled with LSTM layers for activity recognition Number of parameters: 21,990

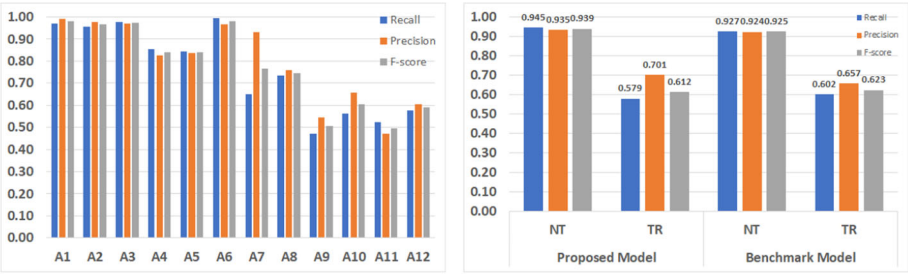

Fig. 3 **a** The performance measures of the benchmark model. **b** Comparison of the performance measures between the proposed model and the benchmark model in terms of classifying non-transitional (NT) and transitional activities (TR)

Table 7 Accuracy of the benchmark model and the proposed model

	Benchmark model	Proposed model
Highest accuracy	0.9054	0.9160
Average accuracy over thirty (30) experiments	0.8893	0.8950
Standard deviation accuracy over thirty (30) experiments	0.0101	0.0096
95% confidence interval	0.8893 ± 0.00363	0.8950 ± 0.00345

classifying transitional activities (A7–A12) at 0.612 compared to 0.623 for the benchmark model. Although the average F-score measure is lower, the average precision is higher, which indicates that the proposed model is more precise in classifying the transitional activities.

Although accepting multiple window segmentations allows our proposed model to achieve better performance, it introduces several challenges. First, the proposed model consumes more memory resources because multiple window segmentations need to be stored. This is also reflected in the dataset preparation for the model training. Since the model needs to capture the dependencies between the multiple window segmentations, a large number of samples (window segmentation) is required to ensure the model generalization. The challenge becomes more complicated when the number of window segmentation is large.

We performed the two independent t-test to determine if there is a statistically significant difference between the two models' accuracy. The proposed and benchmark models are trained thirty (30) times, which is the minimum number of samples for hypothesis tests [34]. Each model's accuracy is recorded, and the average and standard deviation of the models' accuracy are calculated as shown in Table 7. The 95% confidence interval of the model accuracy is also given in the table. The average accuracy of the proposed model is 0.895, which is 0.006 higher than the benchmark model. It is noticed that the standard deviation of the proposed model's accuracy is lower than the benchmark model by 0.0005. The margin of error for a 95% confidence level for the proposed model and benchmark model are 0.00345 and 0.00363 respectively.

We performed two types of hypothesis tests. The first test is to determine if the accuracy of the proposed and benchmark models is equal or not, and the second test is to determine the mean difference of the average accuracy. The significance level of the tests is set to 0.05. The

Table 8 The null hypothesis and the p-value of the hypothesis tests

Null hypothesis	p value
$H_0 : \mu_{proposed\_model} = \mu_{benchmark\_model}$	0.0161
$H_0 : \mu_{proposed\_model} - \mu_{benchmark\_model} > 0.005$	0.5939
$H_0 : \mu_{proposed\_model} - \mu_{benchmark\_model} > 0.007$	0.2943
$H_0 : \mu_{proposed\_model} - \mu_{benchmark\_model} > 0.009$	0.0950
$H_0 : \mu_{proposed\_model} - \mu_{benchmark\_model} > 0.0095$	0.0667

results of the tests are given in Table 8. As can be seen in Table 8, the p-value of the first test is 0.0161, which is lower than the significance level. Therefore, it can be concluded that the average accuracy of both models is not similar. For the second hypothesis test, we performed the test for $\mu = 0.005$, $\mu = 0.007$, $\mu = 0.009$ and $\mu = 0.0095$. As can be seen in Table 8, the p-values of the four tests are above the significance level. Therefore, it is concluded that the mean difference of the average accuracy is about 0.01.

5 Conclusion

In this paper, we propose a deep temporal Conv-LSTM model to model the temporal information of sensor data and activity windows for activity recognition. The proposed model consists of concurrent feature learning pipelines to accept a sequence of activity windows for feature extraction. In addition, the proposed model is integrated with a sequence learning module to learn the temporal features from the concatenated window features. As a result, the proposed model is able to learn a better feature representation of the sensor data for activity recognition. The proposed model is evaluated on a public dataset consisting of dynamic, static and transitional activities, and compared with a benchmark model. The results show that the proposed model performs better than the benchmark model, achieving an accuracy of 0.916, which is 0.013 higher than the accuracy's of the benchmark model. We plan to enhance the network architecture by integrating attention mechanism which can learn the importance of the features to the prediction. The feature learning pipeline can be enhanced by integrating squeeze-and-excitation module to capture more salient features.

Funding This work has been supported in part by the Ministry of Higher Education Malaysia for Fundamental Research Grant Scheme with Project Code: FRGS/1/2019/ICT02/USM/02/1.

Data Availability Not applicable.

Code Availability Not applicable/

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Appendix

Window size: 80

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	890	23	1	0	0	0	4	2	0	0	1	2
A2	31	859	2	1	2	0	3	6	0	0	2	3
A3	4	38	818	0	0	0	2	0	0	0	1	1
A4	0	0	0	918	148	0	2	3	0	3	1	0
A5	4	1	0	150	964	2	2	1	0	0	1	2
A6	0	0	0	0	0	1143	0	1	0	0	1	2
A7	0	0	0	13	14	0	40	1	1	3	5	1
A8	0	0	0	6	16	0	1	36	0	1	0	2
A9	0	0	0	5	3	17	0	2	37	1	31	0
A10	0	0	0	14	0	8	1	1	0	50	0	6
A11	0	0	0	6	2	16	0	0	18	2	42	0
A12	3	0	0	4	7	11	0	6	0	25	0	29

Window size: 100

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	702	34	0	0	0	0	1	1	0	0	1	1
A2	19	696	13	0	0	0	1	0	0	0	1	0
A3	3	5	676	0	0	0	0	0	0	0	0	3
A4	0	0	0	710	139	0	2	0	0	4	2	0
A5	1	1	0	93	800	1	0	1	0	1	4	1
A6	0	0	0	0	0	918	0	0	0	0	2	1
A7	0	1	0	6	10	0	22	1	1	4	15	1
A8	0	1	0	5	11	0	1	33	0	0	0	1
A9	0	0	0	3	1	9	0	2	36	0	24	1
A10	0	0	0	12	0	6	0	0	0	35	1	8
A11	0	1	0	4	1	13	0	1	17	2	30	0
A12	1	2	0	2	3	7	0	2	0	22	0	28

Window size: 120

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	600	2	1	0	1	0	0	1	0	0	1	10
A2	31	562	5	0	1	0	0	2	0	0	0	0
A3	3	11	567	0	0	0	0	0	0	0	1	0
A4	0	0	0	606	99	0	4	1	0	3	0	0
A5	0	1	0	43	697	0	4	3	0	0	3	1
A6	0	0	0	0	0	762	0	0	4	1	1	1
A7	0	0	0	7	12	0	26	1	0	1	5	0
A8	0	0	0	4	7	0	5	24	0	0	0	0
A9	0	0	0	5	3	11	0	0	24	0	21	0
A10	0	0	0	8	0	3	3	0	1	33	0	7
A11	0	0	0	4	2	6	4	0	13	0	28	0
A12	1	0	0	5	3	4	0	2	0	16	0	22

Window size: 140

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	499	21	3	0	0	0	2	1	0	0	0	4
A2	12	484	10	0	1	0	0	10	0	0	0	5
A3	0	5	481	0	0	0	1	0	0	0	0	1
A4	0	0	0	503	98	0	2	3	1	3	0	1
A5	0	0	0	77	552	0	4	8	0	1	3	0
A6	0	0	0	0	0	653	0	0	1	1	0	3
A7	0	0	0	5	12	0	25	0	0	0	1	0
A8	0	0	0	1	3	0	0	28	0	2	0	0
A9	0	0	0	3	0	5	1	2	21	0	23	0
A10	0	0	0	9	0	3	1	0	0	29	0	6
A11	0	0	0	3	1	7	4	0	8	0	28	0
A12	0	0	0	1	2	4	0	0	0	19	0	21

Window size: 128

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	525	49	4	0	0	0	0	1	0	0	0	1
A2	15	446	6	0	0	0	0	1	0	0	0	0
A3	0	10	526	0	0	0	0	0	0	0	1	0
A4	0	0	0	594	68	0	1	4	0	1	0	0
A5	1	1	0	21	674	2	1	2	0	1	5	0
A6	0	0	0	0	0	713	0	0	0	1	1	4
A7	0	0	0	11	6	0	27	2	0	1	2	0
A8	1	0	0	2	5	0	0	32	0	0	0	0
A9	0	0	0	5	1	7	0	0	20	0	25	1
A10	0	0	0	9	2	2	0	0	0	28	0	7
A11	0	0	0	1	3	6	0	1	5	0	39	0
A12	1	0	0	2	2	2	0	2	0	17	0	25

References

1. Abidine BM, Fergani L, Fergani B, Oussalah M (2018) The joint use of sequence features combination and modified weighted SVM for improving daily activity recognition. Pattern Anal Appl 21:119–138. https://doi.org/10.1007/s10044-016-0570-y
2. Tian Y, Zhang J, Wang J et al (2020) Robust human activity recognition using single accelerometer via wavelet energy spectrum features and ensemble feature selection. Syst Sci Control Eng 8:83–96. https://doi.org/10.1080/21642583.2020.1723142
3. Vanrell SR, Milone DH, Rufiner HL et al (2018) Assessment of homomorphic analysis for human activity recognition from acceleration signals. IEEE J Biomed Health Inform 22:1001–1010. https://doi.org/10.1109/JBHI.2017.2722870
4. Ertuğrul ÖF, Kaya Y (2017) Determining the optimal number of body-worn sensors for human activity recognition. Soft Comput 21:5053–5060. https://doi.org/10.1007/s00500-016-2100-7
5. Kanjilal R, Uysal I (2021) The future of human activity recognition: deep learning or feature engineering? Neural Process Lett 53:561–579. https://doi.org/10.1007/s11063-020-10400-x
6. Wang J, Chen Y, Hao S et al (2019) Deep learning for sensor-based activity recognition: a survey. Pattern Recognit Lett 119:3–11. https://doi.org/10.1016/j.patrec.2018.02.010
7. Xu W, Pang Y, Yang Y, Liu Y (2018) Human activity recognition based on convolutional neural network. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp 165–170
8. Bevilacqua A, MacDonald K, Rangarej A et al (2019) Human activity recognition with convolutional neural networks. In: Brefeld U, Curry E, Daly E et al (eds) Machine learning and knowledge discovery in databases. Springer, Cham, pp 541–552
9. Lawal IA, Bano S (2019) Deep human activity recognition using wearable sensors. In: Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments. Association for Computing Machinery, New York, pp 45–48
10. Gil-Martín M, San-Segundo R, Fernández-Martínez F, Ferreiros-López J (2021) Time analysis in human activity recognition. Neural Process Lett 53:4507–4525. https://doi.org/10.1007/s11063-021-10611-w
11. Zhu R, Xiao Z, Cheng M et al (2018) Deep ensemble learning for human activity recognition using smartphone. In: 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), pp 1–5
12. Zehra N, Azeem SH, Farhan M (2021) Human activity recognition through ensemble learning of multiple convolutional neural networks. In: 2021 55th annual Conference on Information Sciences and Systems (CISS), pp 1–5

13. Sikder N, Chowdhury MdS, Arif ASM, Nahid A-A (2019) Human activity recognition using multichannel convolutional neural network. In: 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), pp 560–565
14. Zhang H, Xiao Z, Wang J et al (2020) A novel IoT-perceptive Human Activity Recognition (HAR) approach using multihead convolutional attention. IEEE Internet Things J 7:1072–1080. https://doi.org/10.1109/JIOT.2019.2949715
15. Chen Y, Zhong K, Zhang J et al (2016) LSTM networks for mobile human activity recognition. Atlantis Press, pp 50–53
16. Zebin T, Sperrin M, Peek N, Casson AJ (2018) Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks. In: 2018 40th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp 1–4
17. Guan Y, Plötz T (2017) Ensembles of deep LSTM learners for activity recognition using wearables. Proc ACM Interact Mob Wearable Ubiquitous Technol 1:1–28. https://doi.org/10.1145/3090076
18. Li S, Li C, Li W et al (2018) Smartphone-sensors based activity recognition using IndRNN. In: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers. Association for Computing Machinery, New York, pp 1541–1547
19. Mahmud T, Akash SS, Fattah SA et al (2020) Human activity recognition from multi-modal wearable sensor data using deep multi-stage LSTM architecture based on temporal feature aggregation. In: 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), pp 249–252
20. Ordóñez FJ, Roggen D (2016) Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors. https://doi.org/10.3390/s16010115
21. Mekruksavanich S, Jitpattanakul A (2020) Smartwatch-based human activity recognition using hybrid LSTM network. In: 2020 IEEE SENSORS, pp 1–4
22. Mutegeki R, Han DS (2020) A CNN-LSTM approach to human activity recognition. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp 362–366
23. Li Z, Liu Y, Guo X, Zhang J (2020) Multi-convLSTM neural network for sensor-based human activity recognition. J Phys Conf Ser 1682:012062. https://doi.org/10.1088/1742-6596/1682/1/012062
24. Wang H, Zhao J, Li J et al (2020) Wearable sensor-based human activity recognition using hybrid deep learning techniques. Secur Commun Netw 2020:2132138. https://doi.org/10.1155/2020/2132138
25. Singh SP, Sharma MK, Lay-Ekuakille A et al (2021) Deep ConvLSTM with self-attention for human activity decoding using wearable sensors. IEEE Sens J 21:8575–8582. https://doi.org/10.1109/JSEN.2020.3045135
26. Abdel-Basset M, Hawash H, Chakrabortty RK et al (2021) ST-DeepHAR: deep learning model for human activity recognition in IoHT applications. IEEE Internet Things J 8:4969–4979. https://doi.org/10.1109/JIOT.2020.3033430
27. Xia K, Huang J, Wang H (2020) LSTM-CNN architecture for human activity recognition. IEEE Access 8:56855–56866. https://doi.org/10.1109/ACCESS.2020.2982225
28. Nafea O, Abdul W, Muhammad G, Alsulaiman M (2021) Sensor-based human activity recognition with spatio-temporal deep learning. Sensors. https://doi.org/10.3390/s21062141
29. Xiao Z, Xu X, Xing H et al (2021) A federated learning system with enhanced feature extraction for human activity recognition. Knowl -Based Syst 229:107338. https://doi.org/10.1016/j.knosys.2021.107338
30. Gao W, Zhang L, Teng Q et al (2021) DanHAR: dual attention network for multimodal human activity recognition using wearable sensors. Appl Soft Comput 111:107728. https://doi.org/10.1016/j.asoc.2021.107728
31. Reyes-Ortiz J-L, Oneto L, Sama A et al (2016) Transition-aware human activity recognition using smartphones. Neurocomputing 171:754–767
32. Janidarmian M, Roshan Fekr A, Radecka K, Zilic Z (2017) A Comprehensive analysis on wearable acceleration sensors in human activity recognition. Sensors. https://doi.org/10.3390/s17030529
33. Banos O, Galvez J-M, Damas M et al (2014) Window size impact in human activity recognition. Sensors 14:6474–6499. https://doi.org/10.3390/s140406474
34. Hogg RV, Tanis EA, Zimmerman DL (2010) Probability and statistical inference. Prentice Hall, Upper Saddle River