Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A new CNN-LSTM architecture for activity recognition employing wearable motion sensor data: Enabling diverse feature extraction

Enes Koşar, Billur Barshan [*]

Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, TR-06800 Ankara, Turkey

ARTICLE INFO

Keywords:
Human activity recognition (HAR)
Wearables
Wearable sensors
Motion sensors
Deep learning
Hybrid network models
CNN
LSTM
CNN-LSTM
Feature extraction
Model complexity
UCI HAR dataset
Daily and Sports Activities (DSA) dataset

ABSTRACT

Extracting representative features to recognize human activities through the use of wearables is an area of on-going research. While hand-crafted features and machine learning (ML) techniques have been sufficiently well investigated in the past, the use of deep learning (DL) techniques is the current trend. Specifically, Convolutional Neural Networks (CNNs), Long Short Term Memory Networks (LSTMs), and hybrid models have been investigated. We propose a novel hybrid network architecture to recognize human activities through the use of wearable motion sensors and DL techniques. The LSTM and the 2D CNN branches of the model that run in parallel receive the raw signals and their spectrograms, respectively. We concatenate the features extracted at each branch and use them for activity recognition. We compare the classification performance of the proposed network with three single and three hybrid commonly used network architectures: 1D CNN, 2D CNN, LSTM, standard 1D CNN-LSTM, 1D CNN-LSTM proposed by Ordóñez and Roggen, and an alternative 1D CNN-LSTM model. We tune the hyper-parameters of six of the models using Bayesian optimization and test the models on two publicly available datasets. The comparison between the seven networks is based on four performance metrics and complexity measures. Because of the stochastic nature of DL algorithms, we provide the average values and standard deviations of the performance metrics over ten repetitions of each experiment. The proposed 2D CNN-LSTM architecture achieves the highest average accuracies of 95.66% and 92.95% on the two datasets, which are, respectively, 2.45% and 3.18% above those of the 2D CNN model that ranks the second. This improvement is a consequence of the proposed model enabling the extraction of a broader range of complementary features that comprehensively represent human activities. We evaluate the complexities of the networks in terms of the total number of parameters, model size, training/testing time, and the number of floating point operations (FLOPs). We also compare the results of the proposed network with those of recent related work that use the same datasets.

1. Introduction

Through the pervasiveness of a communicating network of interconnected devices and computing intelligence, wearables have become one of the key elements of the Internet of Things (IoT) ecosystem. Continuous streaming of easily accessible signals, acquired from sensors embedded in wearable devices, provide vast amounts of data that carry valuable information about the user state and well being. Proper processing of these data allows developing innovative solutions to challenging problems (Niknejad et al., 2020).

The aim of human activity recognition (HAR) is to detect and monitor activities automatically through the use of sensory input (Ramanujam et al., 2021; Dhiman and Vishwakarma, 2019). It is pertinent to areas such as ambient intelligence, context-aware systems, assistive technologies, healthcare, biomechanics, sports science, and ergonomics (Gil-Martín et al., 2020; Zhang et al., 2022; Haktanır and Kahraman, 2022; Lattanzi and Freschi, 2020).

Smart environment based and wearable sensor based solutions to HAR exist (Wang et al., 2019). In the former approach, besides the high installation cost and confining the user to a bounded area, the common use of videos, sometimes supplemented with audio signal recordings, entails privacy issues. A more favorable and less costly approach is based on wearable sensor technology which allows direct recording of time-series signals in 3D without any occlusion effects or correspondence problem (Fig. 1). Time-series signals are recorded from multiple sensor axes which can possibly be partitioned into shorter time segments. Accelerometers, gyroscopes, and magnetometers are the commonly used wearable sensor types. Fig. 2 illustrates sample raw signal recordings from the two datasets that we have employed in this study.

One of the challenges of activity recognition, which involves a classification task, is the extraction of features (Chen et al., 2022). Recent studies have favored DL models over ML techniques (Mekruksavanich

[*] Corresponding author.
E-mail addresses: enes@ee.bilkent.edu.tr (E. Koşar), billur@ee.bilkent.edu.tr (B. Barshan).

https://doi.org/10.1016/j.engappai.2023.106529
Received 5 July 2022; Received in revised form 20 April 2023; Accepted 22 May 2023
Available online 28 June 2023

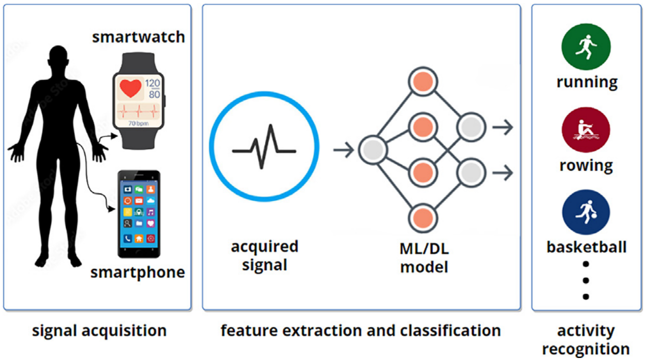


Fig. 1. Human activity recognition through the use of wearable devices.

and Jitpattanakul, 2021; Yen et al., 2020; Xia et al., 2020) since the former has the advantage of providing automated feature extraction and superior classification performance, despite the intensive computational requirements. Specifically, single CNN, single LSTM, and hybrid models that combine CNN and LSTM in different ways are being investigated for HAR. Since DL models learn from data to extract features automatically, generalizable DL models need to be trained on large datasets that contain diverse samples from different classes. Computer vision and natural language processing (NLP) areas are associated with considerably large datasets. Hence, strong vision and language models can be trained that have the capability to extract generalizable features. However, the datasets available in the wearable sensor based HAR area are typically smaller in size. This prevents the DL models to learn diverse features which results in extracting features that are not generalizable for unseen data. Therefore, extracting generalizable features from relatively smaller datasets is a challenging problem for wearable sensor based HAR which enables high classification performance on unseen data.

In this study, we investigate the best DL architecture for extracting highly representative features for the accurate classification of human activities through the use of wearable motion sensors. We propose a new 2D CNN-LSTM hybrid architecture and compare its performance with six existing, commonly used DL models which are 1D CNN, 2D CNN, LSTM, standard 1D CNN-LSTM, 1D CNN-LSTM model proposed in Ordóñez and Roggen (2016), and an alternative 1D CNN-LSTM. We evaluate the seven models based on four performance metrics and their complexities, using two publicly available datasets. Finally, we compare our results with those of recent studies that use the same datasets.

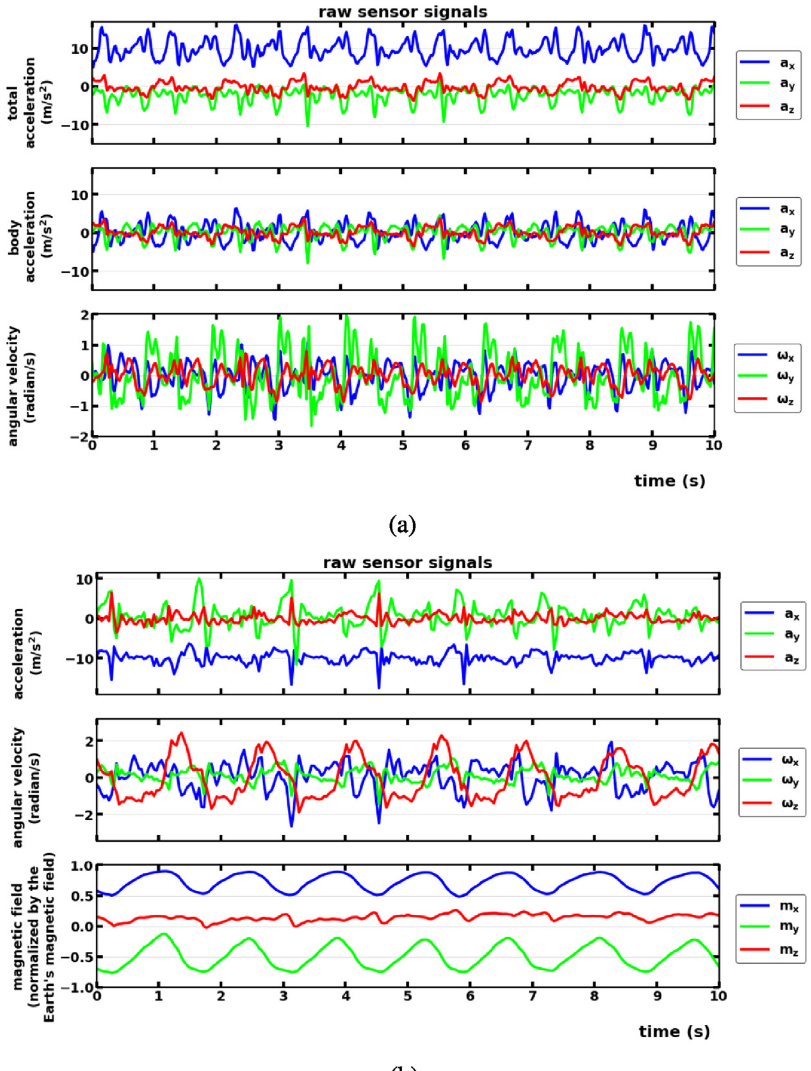
The main contribution of this article is the proposed 2D CNN-LSTM hybrid network architecture which differs from the standard 1D CNN-LSTM hybrid architectures in three respects: First, the proposed network has 2D CNN and LSTM layers in parallel branches instead of sequential (series) form. Second, each branch receives the input signal in a different form as appropriate for the type of layers used in that branch. Third, features extracted in each branch are concatenated and input signals are classified by using the merged features instead of the features extracted from a single model. Although there is a considerable number of works comparing the performances of different network structures, this study focuses on investigating the effect of using several different types of layers on network performance by keeping other factors very similar for each network. Finally, we identify the best way of combining CNN and LSTM layers to create a hybrid model by comparing four hybrid structures, one of which is the proposed one.

The rest of this article is organized as follows: Section 2 reviews the related work employing CNN, LSTM, and hybrid models for HAR through the use of wearable sensor signals. Section 3 starts by presenting the proposed 2D CNN-LSTM hybrid architecture. This is followed by a description of the two publicly available datasets that we have used, preprocessing of the data, resources and implementation of the models, layer and hyper-parameter selection, and the details of the methodology on training, validation, testing, and performance analysis. Section 4 presents the experimental results where we also provide the results of the ablation studies that we have conducted and discuss the results. Finally, we summarize and draw conclusions in Section 5, providing some future research directions.

2. Related work

Since LSTM models provide favorable results on time-series data (Fig. 3(a)), they are a suitable choice for HAR based on wearable sensors (Barut et al., 2020; Chung et al., 2019; Lv et al., 2020; Tufek et al., 2020). CNNs, which are particularly successful models for processing image data, have also been used for HAR (Tufek et al., 2020; Zhu et al., 2019; Sena et al., 2021; Qin et al., 2020) where the acquired signals are typically time-series data. When this is the case, a commonly preferred approach is to use 1D CNNs and perform the convolution only along the time axis (Fig. 3(b)). However, other methods can be also employed such as extracting the spectrogram of the raw signal and feeding it as input to a 1D or 2D CNN (Fig. 3(c)). Although 2D CNNs with spectrogram inputs have been used in processing micro-Doppler radar signals for various purposes (Zhu et al., 2020; Park et al., 2016; Kim and Toomajian, 2016; Kim and Moon, 2016), they are not sufficiently well investigated for wearable sensor signals. To our knowledge, there is a handful of studies where spectrograms of wearable sensor signals are provided as input to a 1D CNN (Ravi et al., 2017; Yao et al., 2017; Pravallika et al., 2020) or to a 2D CNN (Lawal and Bano, 2020; Ito et al., 2018; Li et al., 2020; Pardo et al., 2019).

Besides the single CNN and LSTM models, hybrid structures aiming to combine the strong aspects of both models exist. Hybrid models usually involve a series (cascaded) connection such as CNN-LSTM and LSTM-CNN where the output of the first model is fed as input to the second one in the sequence. Fig. 4 illustrates the standard 1D CNN-LSTM model that we have also implemented in this study for comparison with our proposed model. Some standard 1D CNN-LSTM hybrid model architectures used in previous studies are provided in Table 1. Mekruksavanich and Jitpattanakul (Mekruksavanich and Jitpattanakul, 2021) propose a 1D CNN-LSTM network structure using a four-layer 1D CNN and a single-layer LSTM and show that this structure has superior performance compared to a single LSTM model. However, that study does not compare the performance of the 1D CNN-LSTM hybrid model with CNN models. Studies reported in Mutegeki and Han (2020) and Deep and Zheng (2019) also compare the 1D CNN-LSTM with the LSTM and confirm that using this hybrid model improves the activity classification accuracy. Ordóñez and Roggen (2016) use four 1D CNN and two LSTM layers. The use of two LSTMs in sequence distinguishes it from other studies which usually use only a single LSTM layer. It compares the standard 1D CNN-LSTM (named as DeepConvLSTM in Ordóñez and Roggen (2016)) proposed there with baseline 1D CNN, but not with a single LSTM model. Mekruksavanich and Jitpattanakul (2020) compare the 1D CNN-LSTM with both a single CNN and a single LSTM, verifying that this hybrid model classifies the activities more accurately compared to these two single models. The study reported in Wang et al. (2020) uses three 1D CNN layers and a single recurrent neural network (RNN) layer. Different variations of RNNs are implemented in that study which are LSTM, Gated Recurrent Unit (GRU), and bi-directional LSTM (BiLSTM). GRUs are a gating mechanism in RNNs (Cho et al., 2014) similar to LSTM with a forget gate but has fewer parameters than LSTM since they lack an output gate. BiLSTM is a sequence processing model comprising two LSTMs, one taking the input in the forward and the other in the backward direction. The study compares the results with single 1D CNN and LSTM models. It is shown that while the hybrid model with LSTM layer achieves better accuracy compared to a single 1D CNN and LSTM, hybrid models with GRU and BiLSTM do not improve the single model performances.



(a)

(b)

Fig. 2. Sample raw sensor signals acquired from the (a) UCI HAR Dataset and the (b) DSA Dataset.

Table 1
Existing standard 1D CNN-LSTM hybrid model architectures.

Ref. Mekruksavanich and Jitpattanakul (2021)	Ref. Mutegeki and Han (2020)	Ref. Deep and Zheng (2019)	Ref. Ordóñez and Roggen (2016)	Ref. Mekruksavanich and Jitpattanakul (2020)	Ref. Wang et al. (2020)
1D CNN 1D CNN 1D CNN 1D CNN dropout pooling flatten LSTM dropout dense softmax	1D CNN pooling flatten LSTM softmax	1D CNN 1D CNN dropout pooling flatten LSTM dropout dense softmax	1D CNN 1D CNN 1D CNN 1D CNN flatten dropout LSTM dropout LSTM dropout softmax	1D CNN 1D CNN dropout pooling LSTM dropout softmax	1D CNN pooling 1D CNN pooling 1D CNN pooling LSTM dense batch softmax

As examples of some non-standard models (not included in Table 1), Yao et al. (2017) propose a hybrid architecture composed of GRU and CNN networks. The proposed architecture outperforms four ML algorithms and three variants of the proposed method. Hamad et al. (2020) use 1D CNN and LSTM separately to extract features from raw data which are then concatenated. Their study shows that using two different models enables better feature extraction compared to employing two models of the same type (both being LSTMs or both being 1D CNNs) for the same purpose. Huynh-The et al. (2021) combine CNN-extracted features with hand-crafted features to improve the activity recognition performance. In Peng et al. (2018), the authors employ a LSTM layer following a CNN layer to recognize complex activities while they use a single CNN model for simple activities. Mukherjee et al. (2020) use majority voting over three different DL models to classify the activities. Most of the previous studies have compared hybrid models with either single CNN or single LSTM models (but not both) despite that a comparison with both would have been more comprehensive in

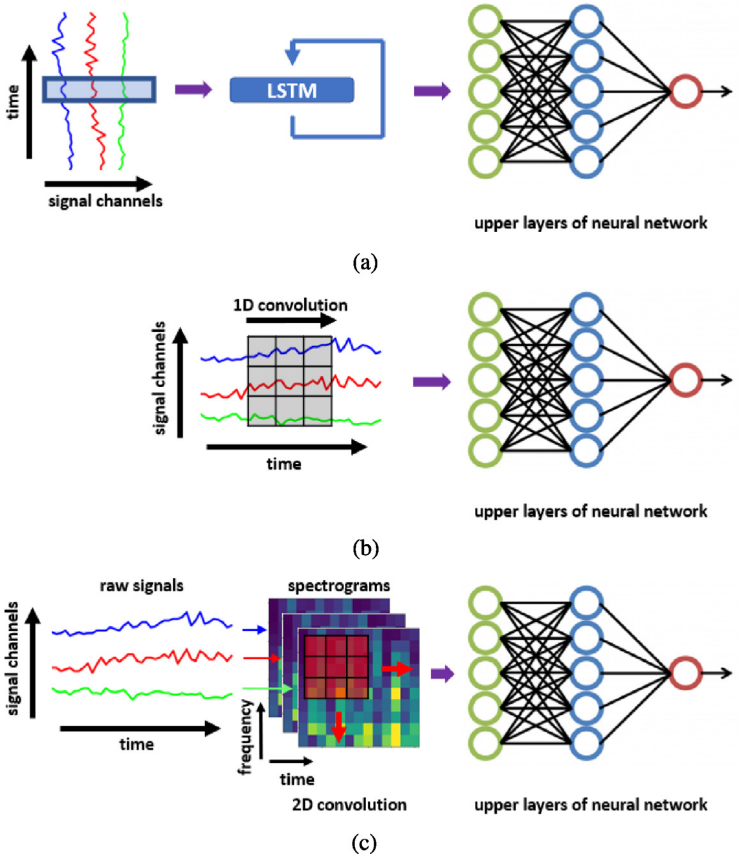


Fig. 3. Single network models. (a) The LSTM model and the (b) 1D CNN model processing the raw input signals, and the (c) 2D CNN model processing the spectrogram of the raw input signals.

verifying the superiority of hybrid models. Furthermore, investigating different ways of combining CNN and LSTM networks could enhance the performance even further. However, this issue is not considered at all in the above-mentioned studies.

There are also studies that propose some novelty at the layer level rather than the model architecture level. Tang et al. (2023) propose to use the hierarchical-split idea for CNN layers that can improve the ability to represent features at multiple scales by capturing a broader range of receptive fields associated with human activities in a single feature layer. Another study (Han et al., 2022) proposes a novel approach that uses heterogeneous convolution inspired by grouped convolution to enhance the performance of activity recognition without increasing the computational overhead. Authors of Huang et al. (2022a) propose a new type of CNN that leverages filter activation to activate the seemingly unimportant filters from the perspective of enhancing accuracy. The proposed approach requires only a single network instead of multiple networks to be deployed on resource-limited embedded devices. To address the issue of sequential weakly labeled multi-activity recognition and localization, Wang et al. (2021) propose a recurrent attention network (RAN) that iteratively applies attention to multiple activities within a single sample. By doing so, the RAN effectively reduces the need for manual labeling and annotation. Huang et al. (2022b) propose Channel Equalization (CE) as a solution to the 'channel collapse' problem where most channels do not contribute much information and only a few are relied on. CE tackles this by activating all channels via a whitening or decorrelation operation.

3. Methodology

3.1. The proposed model

We propose a hybrid model where a 2D CNN and a LSTM network are combined in parallel, as illustrated in Fig. 5. Since LSTMs are favored for time-series data and CNNs are more suitable for processing image data, we feed the LSTM branch with the raw time-series sensor readings and the CNN branch with the spectrogram of the raw sensor recordings. While LSTM focuses on the time-dependent patterns in the data, CNN considers the frequency patterns as well, enabling each model to extract features representing different characteristics of the signal that convey complementary information. By merging such features, we can obtain better feature representation of the signal, resulting in improved activity classification performance.

3.2. Datasets

In this subsection, we briefly describe the two publicly available datasets that we have employed in this study.

3.2.1. UCI HAR dataset

Thirty participants between the ages 19 and 48 perform six activities while the accelerometer and gyroscope sensors of a smartphone (Samsung Galaxy S2), carried on their waist, records the data (Reyes-Ortiz et al., 2015; Anguita et al., 2013; Reyes-Ortiz et al., 2016). Recorded activities are walking on a flat surface, walking upstairs, walking downstairs, sitting, standing, and laying. The number of samples in

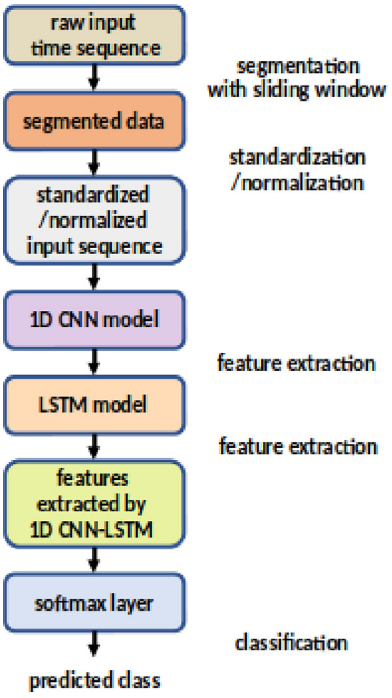


Fig. 4. The standard 1D CNN-LSTM network architecture.

each activity class are comparable but not equal, causing this dataset to be slightly imbalanced. The accelerometer captures tri-axial linear acceleration while the gyroscope detects tri-axial angular velocity (or rate) at a sampling frequency of $f_s = 50$ Hz. Both the total acceleration and the body component of acceleration are provided in this dataset. Body components are obtained by subtracting the gravitational component from the total acceleration. Signals are segmented by using an overlapping sliding window with a length of 128 readings/window. Each time segment contains data from nine channels in parallel: x, y, and z axes of (i) accelerometer total components, (ii) accelerometer body components, and (iii) gyroscope angular velocity components. The acquired dataset has been randomly partitioned into two, where data from 70% of the subjects are used for training and data from the remaining 30% are employed for testing.

3.2.2. Daily and sports activities (DSA) dataset

Eight subjects (four male, four female) between the ages 20 and 30 perform 19 different daily and sports activities (Altun and Barshan, 2019, 2013). During the experiments, the subjects wear five sensor units placed on their chest, right/left arm, and right/left leg. Each sensor unit contains three tri-axial sensors: accelerometer, gyroscope, and magnetometer. These three sensor types capture signals in the x, y, and z axes at a sampling frequency of $f_s = 25$ Hz. Hence, there are 45 signal channels (5 sensor units × 3 sensor types × 3 axes). The subjects perform each activity for five minutes after which the recorded signals are divided into non-overlapping 5-sec segments. The activity classes are balanced since there is an equal number of samples from each activity type.

It is notable that the UCI HAR Dataset contains a small number of activities (six) performed by a large number of participants (30) whereas the DSA Dataset contains a large number of activities (19) performed by a smaller number of participants (eight). The UCI HAR Dataset contains 269 MB data while the DSA Dataset comprises 402 MB data. Given the smaller number of participants of the DSA dataset, resulting in less variation in the acquired data, and the larger number of activities it contains which need to be differentiated, this dataset is more challenging in terms of learning the discriminatory characteristics of the different activities compared to the UCI-HAR Dataset.

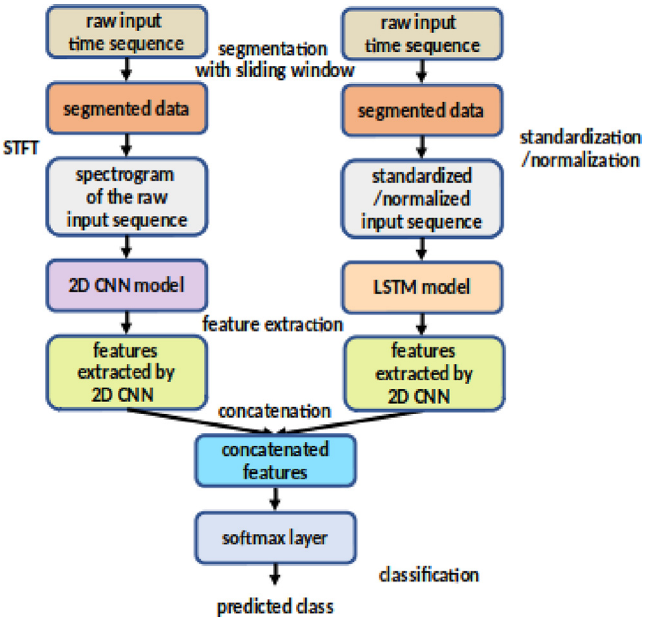

raw input time sequence
segmentation with sliding window
raw input time sequence
segmented data
segmented data
STFT
standardization /normalization
spectrogram of the raw input sequence
standardized /normalized input sequence
2D CNN model
LSTM model
feature extraction
features extracted by 2D CNN
features extracted by 2D CNN
concatenation
concatenated features
softmax layer
classification
predicted class

Fig. 5. The proposed 2D CNN-LSTM hybrid architecture.

3.3. Preprocessing

3.3.1. Preprocessing the data for the 1D CNN and LSTM branches of the DL models

Since each sensor type provides output in different ranges with different units, the signals can be normalized or standardized. Mekruksavanich and Jitpattanakul (2021) employ standardization for the UCI HAR Dataset and Yurtman and Barshan (2017), Yurtman et al. (2018), Barshan and Yurtman (2020), Yurtman et al. (2021) employ normalization for the DSA Dataset. For compatibility with these related studies, we standardized the signals in the UCI HAR Dataset and normalized those in the DSA Dataset, considering each signal channel separately.

We standardized the UCI HAR Dataset as follows:

$$x[n]_{\text{standardized}} = \frac{x[n] - \bar{x}[n]}{\sigma_x}$$

where $x[n]$ is the data sequence in one time segment of data, $\bar{x}[n]$ is the average value of $x[n]$, and σ_x is the standard deviation of $x[n]$. Standardized sequences have zero mean and unit variance.

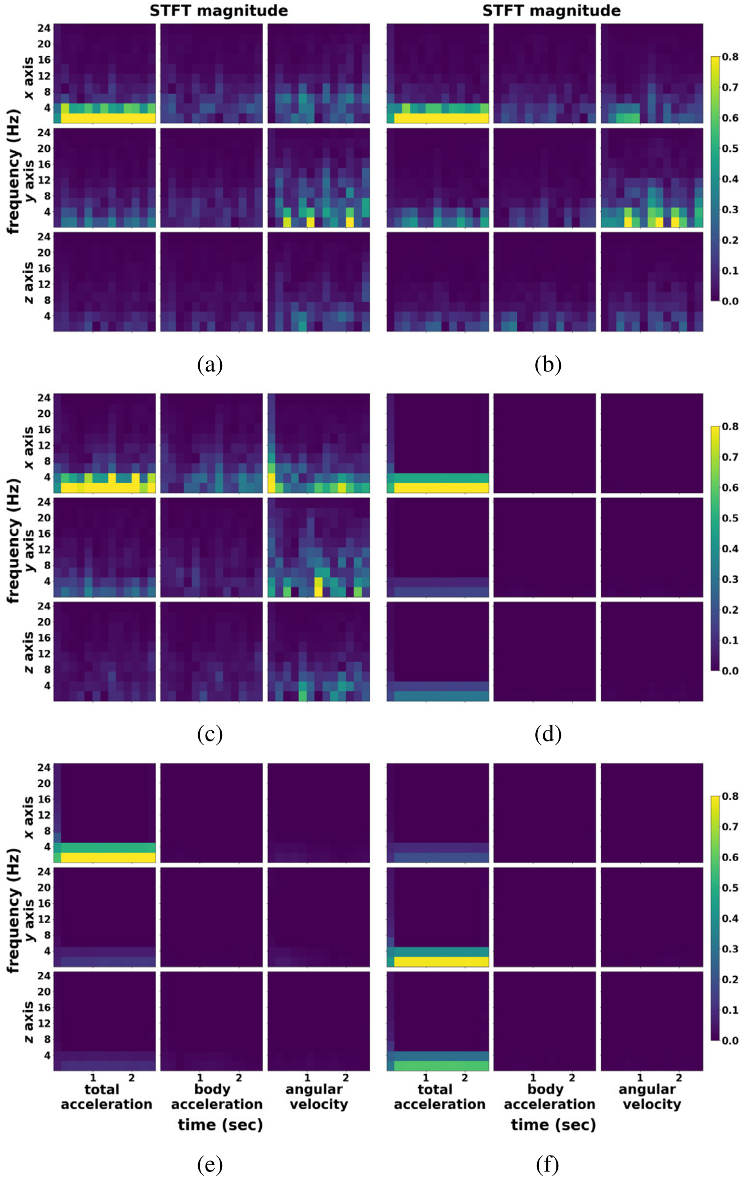
For the DSA Dataset, we modified the normalization formula by subtracting the average value of the sequence from the sequence instead of its minimum value. Thus, we normalized the data using the following equation:

$$x[n]_{\text{normalized}} = \frac{x[n] - \bar{x}[n]}{x_{\max} - x_{\min}}$$

Here, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the sequence $x[n]$, respectively.

3.3.2. Preprocessing the data for the 2D CNN branches of the DL models

We did not standardize/normalize the data for the 2D CNN because these operations remove the zero frequency (DC) component of the signal, which is not desirable. We directly convert the data in each time segment of each signal channel into a spectrogram by taking the magnitude of its short-time Fourier transform (STFT) (Oppenheim et al., 1999). To do this, we first padded the beginning and the end of each time segment with zeroes to obtain a total of 140 samples per time



(a) (b)

(c) (d)

(e) (f)

Fig. 6. Sample spectrograms obtained from the UCI HAR Dataset for the activities of (a) walking on a flat surface, (b) walking upstairs, (c) walking downstairs, (d) sitting, (e) standing, and (f) laying.

segment for both datasets. Based on the 140 samples, we have obtained 14 subsegments of 20 samples each, with 50% overlap between consecutive subsegments. We took the Discrete Fourier Transform (DFT) of each subsegment to get 11 frequency components in the interval $[0, \frac{f_s}{2}]$ with frequency increments of $\frac{f_s}{20}$ where f_s is the sampling frequency. As a result, the spectrogram of each time segment has dimensionality 11 × 14 for both datasets. Thus, the time segments of the two datasets with dimensionalities 128 × 9 and 125 × 45 are respectively transformed to spectrograms with dimensionalities 11 × 14 × 9 and 11 × 14 × 45. Sample spectrograms obtained from the two datasets are provided in Figs. 6 and 7.

3.4. Implementation and resources

We have implemented all seven models using the Keras and Tensorflow libraries in Python. Since tuning the hyper-parameters and processing the datasets are computationally intensive, we used the GPUs of two different platforms to expedite the computations. These are the Amazon Web Services (AWS) platform with EC2 p2.xlarge GPU for tuning the hyper-parameters and the Google Colab platform that provides 12 GB NVIDIA Tesla K80 GPU for testing the final models. We used Intel(R) Core(TM) i7-10510U CPU while measuring the testing times.

3.5. Layer selection

Layers can be selected in two ways while making a performance comparison between the different DL models. The first option is to optimize the number of each layer type for each model. While this approach seems to be optimal, it brings an uncertainty about whether the performance differences among the different models are caused by different layer types or different numbers of layers. The alternative is to fix the type and the number of layers for each model and only change the type of a single layer as appropriate to each model. In this approach, performance differences are only the result of the different layer type employed in only one layer of each model. To be able to test our proposed model in both situations, we used the UCI HAR Dataset for the first case and the DSA Dataset for the second case.

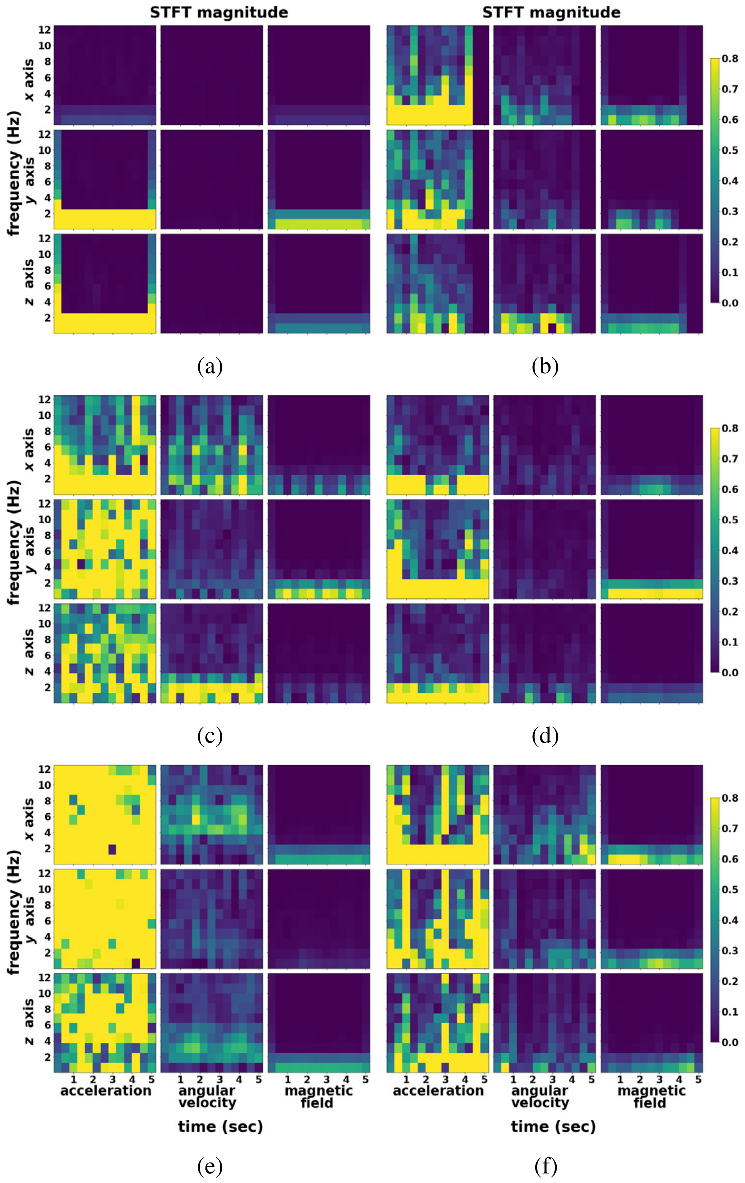


Fig. 7. Sample spectrograms obtained from the DSA Dataset for the activities of (a) lying on the right side, (b) ascending stairs, (c) walking on a treadmill at a speed of 4 km/h in flat position, (d) rowing, (e) jumping, and (f) playing basketball.

In processing the UCI HAR Dataset, we optimized the number of layers in each type of network before selecting the other hyper-parameters. In these experiments, we observed the effect of changing the number of layers while keeping the other hyper-parameter values fixed. Model layers are displayed in Figs. 8 and 9. We obtained the optimum number of layers as four for 1D CNN, two for 2D CNN, three for LSTM, and three for the standard 1D CNN-LSTM. The optimal numbers of dense (fully connected) layers are one, two, one, and one, respectively. Since using a pooling layer did not improve the performance of 1D CNN and 2D CNN models, we did not use pooling layers in the networks developed in this study. Because the alternative 1D CNN-LSTM and the proposed 2D CNN-LSTM model are composed of single networks whose layers are already optimized as described above, we did not conduct a separate layer selection process for them. We used dropout layers as regularizers in between two layers which are placed before the dense layers in all seven models.

In processing the DSA Dataset, we fixed the total number of layers in each model and only changed the type of the first layer. The first layers used in the respective models are 1D CNN, 2D CNN, LSTM, 1D CNN and LSTM in series, 1D CNN and LSTM in parallel, 2D CNN and LSTM in parallel. Model layers are illustrated in Figs. 10 and 11. Since the remaining layers of the networks are identical, differences in the results with this dataset are the consequence of including different types of first layers.

3.6. Hyper-parameter selection

After determining the number of layers in each type of network, we need to select the other hyper-parameters. Although hyper-parameter selection is an on-going research area, there are some common practices, one of which is grid searching. However, when the number of hyper-parameters is large, the time required for grid searching grows exponentially. Finding the optimal hyper-parameters through this computationally intensive process takes a long time.

Another approach is selecting the hyper-parameters sequentially or one at a time. In this method, the value of only one hyper-parameter is changed over an interval. After the best value is selected for that hyper-parameter, it is kept fixed. Then, the next hyper-parameter is tuned in a pre-determined interval. Since many hyper-parameter combinations

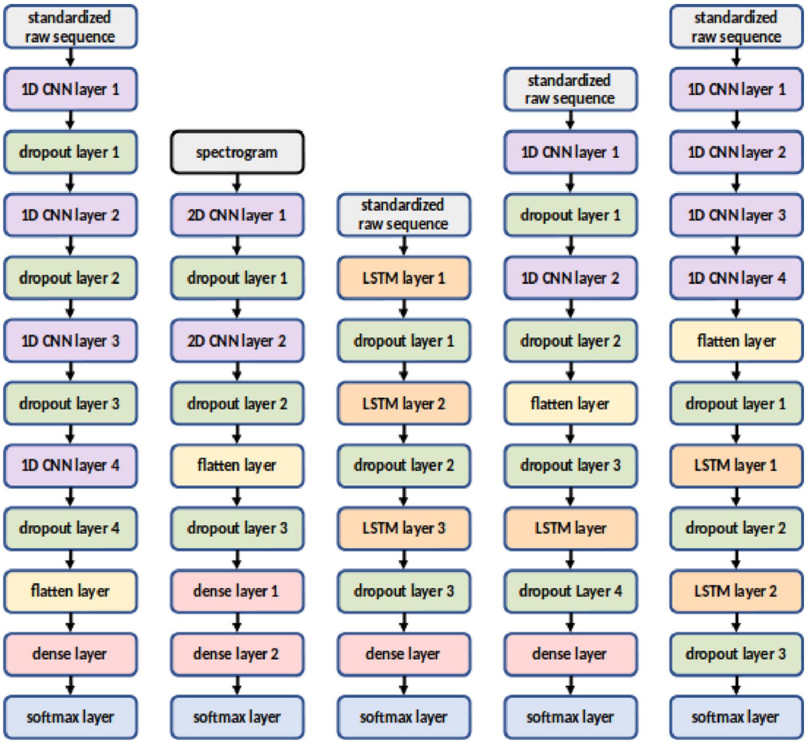


Fig. 8. Series structure and layers of the 1D CNN, 2D CNN, LSTM, standard 1D CNN-LSTM, and the Ordóñez and Roggen 1D CNN-LSTM models used in processing the UCI HAR Dataset.

are not considered, this method is not optimal but computationally less intensive.

The third method is Bayesian optimization (Snoek et al., 2012) used in Mekruksavanich and Jitpattanakul (2021) for HAR. This is the method that we have adopted for hyper-parameter selection. We have used Scikit-Optimize library in Python for the necessary implementation tools. We have employed EC2 p2.xlarge GPU instance in the AWS platform and the “gbrt_minimize” function from Scikit-Optimize library for hyper-parameter tuning using Bayesian optimization. Before the optimization process, the set of model hyper-parameters to be optimized and their initial values are provided as input to the function. During the optimization process, based on the model performance for the previously selected hyper-parameter values, a new set of hyper-parameter values is automatically selected by the algorithm to improve the model performance. The total number of repetitions for this process is chosen as 50 for each model, taking into account the time and the computational resources. Thus, we have considered 50 hyper-parameter combinations for each model to determine the optimum set of hyper-parameter values that provide the best performance for that model.

For the UCI HAR Dataset, the number of epochs was set to 50. With the L1SO cross validation we used for the DSA Dataset (see Section 3.7), we conducted 20 epochs per subject, resulting in a total of 160 epochs $(= 8 \text{ subjects} \times 20 \text{ epochs/subject})$.

There are some common practices on the choice of some of the hyper-parameter values for the network layers. Among these hyper-parameters are the optimizer, loss function, and the activation function which we have selected directly, without tuning. We have employed *Adam* as the optimizer and *Categorical Cross Entropy* as the loss function for all models. We have used the sigmoid activation function for the dense layers and the ReLU activation function for the 1D CNN layers of the models developed for processing both datasets, as well as the 2D CNN layers used for processing the UCI HAR Dataset. However, excluding activation functions in the 2D CNN layers while processing the DSA Dataset resulted in better classification accuracy. Therefore, we did not use any activation function for those layers. Also, the last dense layers of six of the networks developed for processing the DSA Dataset do not contain any activation functions.

3.7. Training, validation, testing methods, and performance analysis

Ideally, a large dataset should be divided into three parts or three separate datasets should be used for the training and testing (inference) process of a DL model: training, validation, and test datasets. Training and validation sets should be used in tuning the hyper-parameters to select their optimal values. Then, the final model should be trained on the union of the training and validation sets and the final performance analysis should be done on the test set. Also, each participant’s data should be included in only one of the three sets (training, validation, test sets) to eliminate any bias during the performance analysis. If there is any overlap in the data used during training, validation, and testing, this would not be a fair evaluation since the models would be tested on data similar to what they were trained on. Furthermore, in real-world applications, it is likely for a DL model to encounter new (unseen) data, different than those used for training.

The UCI HAR Dataset is originally partitioned into training and test sets as follows: Of the 30 subjects, data from subjects 2, 4, 9, 10, 12, 13, 18, 20, and 24 are assigned as test data whereas data from the remaining subjects are used for training. Since a validation set is not provided, we partitioned the data acquired from the 21 training subjects into two. We used data from subjects 1, 3, 5, 6, 7, 8, 11, 14, 15, 16, 17, 19, 21, 22, 23, 25 for training and those from subjects 26, 27, 28, 29, and 30 for validation. While selecting the hyper-parameters, we trained the model on the training set and validated the results for the given set of hyper-parameters on the validation set. After this step, we determine the hyper-parameter values performing the best on the validation set as the optimum ones which are then used in building the

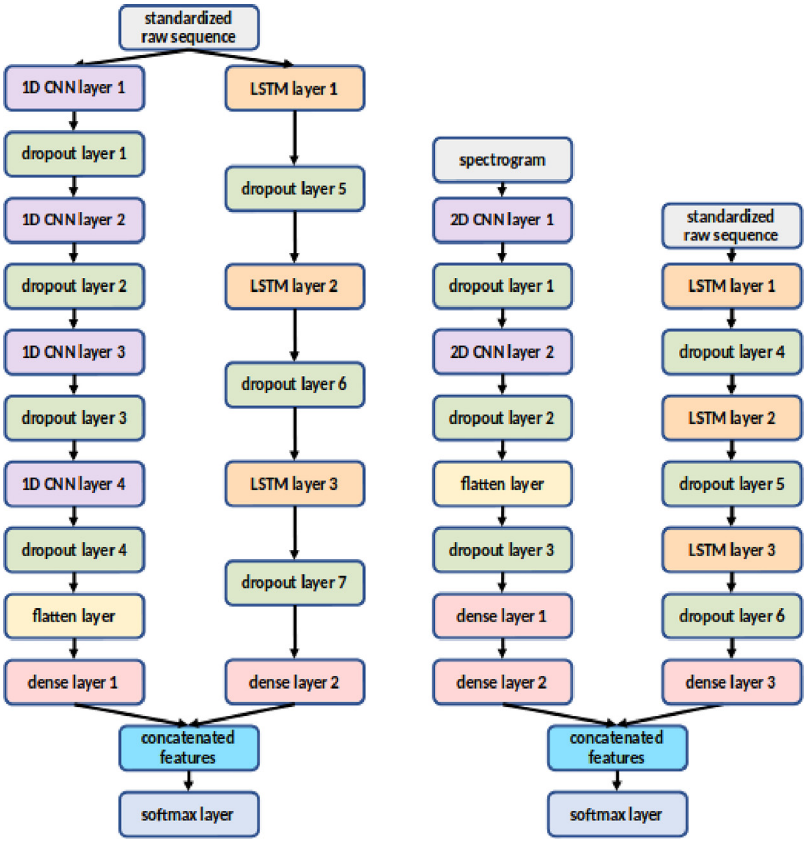


Fig. 9. Parallel structure and layers of the alternative 1D CNN-LSTM and the proposed 2D CNN-LSTM models used in processing the UCI HAR Dataset.

final model. We train the final model on the union of the training and validation sets and evaluate its performance on the test set.

Accuracy is one of the commonly used performance measures. However, when the activity classes are not balanced, F_1 score, recall, and precision measures are more appropriate for evaluating the performance. In this study, we used all four performance metrics when processing the UCI HAR Dataset, which is somewhat imbalanced, and employed the accuracy metric for the DSA Dataset which is balanced in that it contains an equal number of samples from each activity class.

When the amount of data is limited, cross-validation techniques can be employed. For processing the DSA Dataset in this study, we employ leave-one-subject-out cross validation (L1SO) which uses all the data except one subject's data for training and tests the trained model with the data from the left-out subject. The process is repeated for all subjects and the average value of the performance metric is calculated over the subjects. L1SO method is preferred to the multi-fold cross-validation technique where each subject's data are distributed randomly to both the training and the test set, resulting in a bias against the test set. Consequently, the performance measure does not reflect the actual performance of the model on real data. We used L1SO cross validation for both hyper-parameter selection and testing the final models when processing the DSA Dataset because of its smaller number of subjects.

Since DL models are stochastic, single execution of an experiment can provide misleading results. Therefore, repeating each experiment multiple times (10 times in this study) and reporting the average and standard deviation values would be more appropriate. This is done for the L1SO cross validation used with the DSA Dataset as well where the whole cross-validation process was repeated 10 times.

4. Experimental results and discussion

4.1. Hyper-parameter optimization results

The tuned hyper-parameter values for the proposed 2D CNN-LSTM hybrid model are provided in Tables 2 and 3 for the two datasets. Those for six of the other models that we have implemented in this comparative study are provided in Tables A.1–A.10 of Appendix. We note that we did not tune the parameters of Ordóñez and Roggen's 1D CNN-LSTM model and used the hyper-parameters already provided in their paper (Ordóñez and Roggen, 2016). This is because, like the standard 1D CNN-LSTM model, this model comprises a series combination of 1D CNN and LSTM single models and we have already optimized the parameters of the standard 1D CNN-LSTM model.

Total times spent for tuning the hyper-parameters of the models on both datasets are displayed in Tables 4 and 5. Tuning the hyper-parameters of the models is a computationally intensive process that took about 29.79 h for the UCI HAR Dataset and 25.53 h for the DSA Dataset, totaling to 55.32 h (for six of the models). Note that hyper-parameter tuning needs to be done only once and the tuned hyper-parameters can be used in (near) real-time applications without tuning at each run of the DL algorithms.

4.2. Performance comparison of the proposed model with the six existing models

The results of our experiments on the two datasets are tabulated in Tables 6 and 7. Since the imbalance of the UCI HAR Dataset is not too significant, the four performance metrics for each model are comparable for this dataset. The accuracy performance metric values based on the UCI HAR Dataset (Table 6) are consistently higher than those based on the DSA Dataset (Table 7) by 2.71–11.31%. This is mainly because there are only six activities to recognize in the UCI HAR

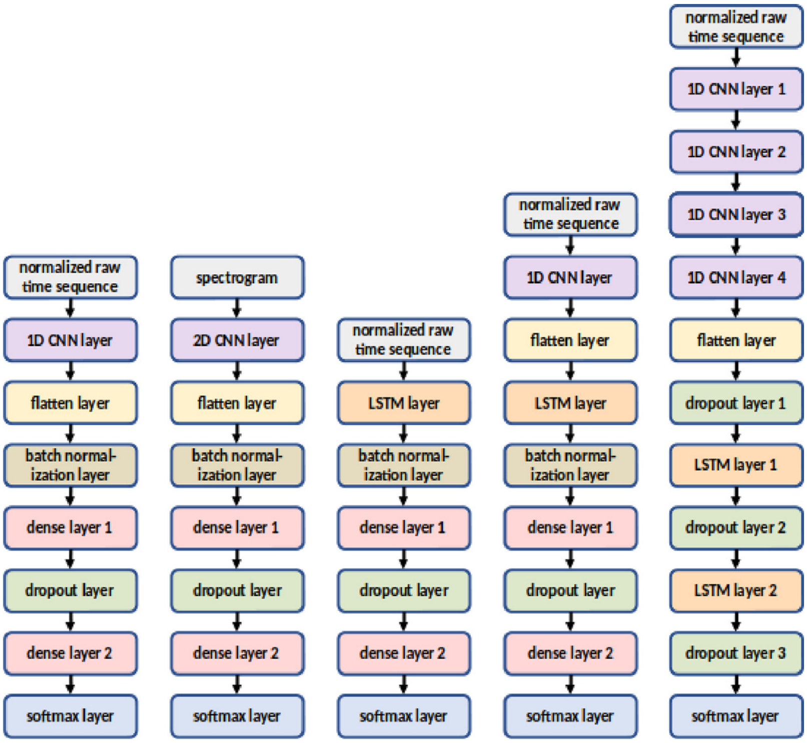


Fig. 10. Series structure and layers of the 1D CNN, 2D CNN, LSTM, standard 1D CNN-LSTM, and the Ordóñez and Roggen 1D CNN-LSTM models used in processing the DSA Dataset.

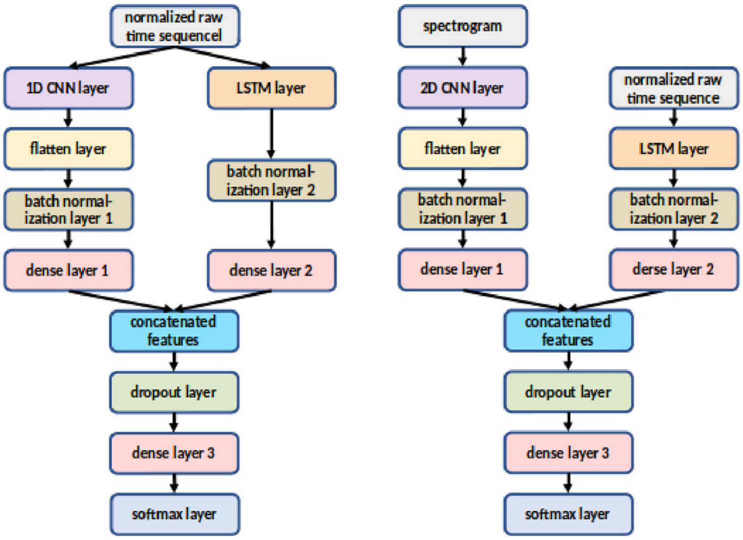


Fig. 11. Parallel structure and layers of the alternative 1D CNN-LSTM and the proposed 2D CNN-LSTM models used in processing the DSA Dataset.

Dataset whereas there are 19 in the DSA Dataset. The standard deviations in the two tables are comparable.

The results indicate that the proposed 2D CNN-LSTM hybrid model achieves better accuracy, F_1, recall, and precision scores compared to the six existing models, namely, 1D CNN, 2D CNN, LSTM, standard 1D CNN-LSTM, 1D CNN-LSTM model proposed in Ordóñez and Roggen (2016), and the alternative 1D CNN-LSTM. The proposed hybrid network achieves an average accuracy of 95.66% on the UCI HAR Dataset and 92.95% on the DSA Dataset. These figures are, respectively, 2.45% and 3.18% above the accuracy of the single 2D CNN model that ranks the second in all performance metrics. In fact, the proposed 2D CNN-LSTM hybrid model outperforms both of its individual components (the single 2D CNN and the single LSTM models) considerably for both datasets. For the UCI HAR Dataset, the four performance metrics improve by 1.98–2.52% and by 4.89–5.63% compared to the single 2D CNN and the single LSTM models, respectively. On the other hand, for the DSA Dataset, the accuracy metric improves by 3.18% with respect to the single 2D CNN and by 14.23% compared to the single LSTM.

When we compare the results of the standard 1D CNN-LSTM model with its individual components (the single 1D CNN and LSTM), we

Table 2
Optimized hyper-parameters of the proposed 2D CNN-LSTM in processing the UCI HAR Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer 1 filter number	64	[4, 256]
CNN layer 2 filter number	64	[4, 256]
CNN layer 1 filter width	3	[3, 7]
CNN layer 2 filter width	3	[3, 7]
LSTM layer 1 neuron number	64	[8, 512]
LSTM layer 2 neuron number	64	[8, 512]
LSTM layer 3 neuron number	64	[8, 512]
dense layer 1 neuron number	64	[8, 512]
dense layer 2 neuron number	64	[8, 512]
dense layer 3 neuron number	64	[8, 512]
dropout layer 1 probability	0.5	[0.1, 0.9]
dropout layer 2 probability	0.5	[0.1, 0.9]
dropout layer 3 probability	0.5	[0.1, 0.9]
dropout layer 4 probability	0.5	[0.1, 0.9]
dropout layer 5 probability	0.5	[0.1, 0.9]
dropout layer 6 probability	0.5	[0.1, 0.9]
learning rate	0.001	[10^{-6}, 0.1]

Table 3
Optimized hyper-parameters of the proposed 2D CNN-LSTM in processing the DSA Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer filter number	103	[4, 256]
CNN layer filter width	5	[3, 7]
LSTM layer neuron number	297	[8, 512]
dense layer 1 neuron number	197	[8, 512]
dense layer 2 neuron number	144	[8, 512]
dense layer 3 neuron number	399	[8, 512]
dropout layer probability	0.76497736	[0.1, 0.9]
learning rate	4.16143569×10^{-5}	[10^{-6}, 0.1]

Table 4
Total hyper-parameter tuning times for the UCI HAR Dataset for 50 hyper-parameter combinations and 50 epochs.

Model name	Total time (s)
1D CNN	4,438
2D CNN	3,659
LSTM	30,911
standard 1D CNN-LSTM	3,740
alternative 1D CNN-LSTM	32,563
proposed 2D CNN-LSTM	31,926

Table 5
Total hyper-parameter tuning times for the DSA dataset for 50 hyper-parameter combinations and 160 epochs.

Model name	Total time (s)
1D CNN	7,464
2D CNN	4,141
LSTM	11,324
standard 1D CNN-LSTM	10,621
alternative 1D CNN-LSTM	41,724
proposed 2D CNN-LSTM	16,649

do not observe a similar performance boost. This is mainly because of the inferior performance of the standard 1D CNN-LSTM network compared to the proposed model. The four performance metrics of the standard 1D CNN-LSTM model are below those of the proposed 2D CNN-LSTM hybrid network by 4.04–4.45% for the UCI HAR Dataset whereas its accuracy metric is lower than that of the proposed 2D CNN-LSTM model by 8.67% when the DSA Dataset is processed. The different performances of the two hybrid networks stem from the differences in their network architecture as well as the type of input that they receive and process. The standard 1D CNN-LSTM architecture uses the 1D CNN and LSTM networks consecutively to extract features from only one kind of input signal (the raw signals) while the proposed model employs the 2D CNN and LSTM in separate branches in parallel, each receiving a different form of input (spectrogram and raw signals, respectively).

To determine whether the superiority of the proposed 2D CNN-LSTM network model arises from the use of two different types of input or from its parallel architecture, we also compare it with an alternative 1D CNN-LSTM network (Hamad et al., 2020), which has a parallel structure similar to that of the proposed model. In both the proposed model and the alternative 1D CNN-LSTM network, the parallel running CNN and LSTM subnetworks extract their own features which are then merged and classified with a softmax layer. The main difference is that while the proposed model takes input signals in two different forms, the alternative 1D CNN-LSTM hybrid model receives the raw signals as input for both the 1D CNN and LSTM branches of the network. The results in Tables 6 and 7 indicate that the performance metrics of the alternative 1D CNN-LSTM model are consistently lower than those of the proposed 2D CNN-LSTM model by 4.47–5.26% for the UCI HAR Dataset and by 8.35% for the DSA Dataset. The performance of the alternative 1D CNN-LSTM is comparable to the performances of its two individual components. More specifically, the four performance metrics of the alternative 1D CNN-LSTM in Table 6 consistently lie in between the individual performance metrics of the single 1D CNN and single LSTM models. This is also the case in Table 7, although the accuracy figure is much closer to that of 1D CNN. Since the alternative 1D CNN-LSTM model performance metrics are always lower than those of the 1D CNN, and consistently with larger standard deviation values, between these two models, it would be preferable to use the 1D CNN model which requires less training time.

The performance comparison between the proposed 2D CNN-LSTM and the alternative 1D CNN-LSTM models reveals that using inputs of different nature (raw signals and spectrogram) enables the proposed model to extract features that enhance and complement each other whereas using the same form of input in both branches of the network causes the alternative 1D CNN-LSTM model to extract possibly overlapping or redundant features. Hence, the proposed model succeeds in extracting features with better representation of activities, resulting in improved activity recognition performance.

The results indicate that the use of spectrograms has a particularly positive effect on DL model performance. We note that the single 2D CNN, also using the spectrogram of the raw signals as input, exhibits the second best performance on both datasets. The use of two inherently different subnetwork models (2D CNN and LSTM) that run in parallel also contributes to the superiority of the proposed model.

4.3. *Loss versus epoch, accuracy versus epoch graphs and confusion matrices of the proposed model*

Figs. 12–14 show the loss versus epoch plot, accuracy versus epoch plot, and the confusion matrix for the proposed 2D CNN-LSTM hybrid model processing the UCI HAR Dataset. The corresponding results for the DSA Dataset are provided in Figs. 15–17.

Fig. 12 indicates that the proposed model for the UCI HAR Dataset works properly since the loss decreases gradually during both training and testing. In Fig. 13, we observe that there is no overfitting issue since there is a small gap between the training and test accuracies. This is achieved by using dropout layers for regularization.

Fig. 15 illustrates that the loss decreases quite smoothly which is an indicator of a good learning rate for the proposed model used in processing the DSA Dataset. On the other hand, this model has a very large dropout probability of 0.76 which means that the model does not use most of the neurons during the training phase while it uses all of them for testing. Therefore, we observe in Fig. 16 that the test accuracy is above the training accuracy until the last few epochs. Then, training and test accuracies converge approximately to the same value. This is a good indicator for the generalizability of the model. The large dropout

Table 6
Comparison of the implemented single and hybrid models in terms of their subnetworks, merging method, and performance metrics. Average values of accuracy, F_1 score, recall, and precision metrics plus/minus one standard deviation are provided for processing the UCI HAR Dataset.

	Model name	Subnetworks	Merging method	Input types	Accuracy ± std (%)	F_1 Score ± std (%)	Recall ± std (%)	Precision ± std (%)
Single models	1D CNN	N/A	N/A	raw input	90.93 ± 0.50	91.23 ± 0.49	91.17 ± 0.48	91.47 ± 0.48
	2D CNN	N/A	N/A	spectrogram	93.21 ± 1.60	93.18 ± 1.65	93.15 ± 1.64	93.67 ± 1.41
	LSTM	N/A	N/A	raw input	90.03 ± 0.68	90.41 ± 0.69	90.30 ± 0.69	90.76 ± 0.66
Hybrid models	standard 1D CNN-LSTM	1D CNN LSTM	series	raw input	91.21 ± 0.57	91.43 ± 0.58	91.39 ± 0.60	91.61 ± 0.52
	Ordóñez-Roggen 1D CNN-LSTM	1D CNN LSTM	series	raw input	87.12 ± 0.98	87.61 ± 0.89	87.48 ± 0.96	88.43 ± 0.00
	alternative 1D CNN-LSTM	1D CNN LSTM	parallel	raw input + raw input	90.40 ± 0.79	90.77 ± 0.73	90.66 ± 0.78	91.18 ± 0.60
	proposed 2D CNN-LSTM	2D CNN LSTM	parallel	spectrogram + raw input	95.66 ± 0.63	95.62 ± 0.65	95.67 ± 0.63	95.65 ± 0.63

Table 7
Comparison of the implemented single and hybrid models in terms of their subnetworks, merging method, and accuracy performance metric. Average value of accuracy plus/minus one standard deviation is provided for processing the DSA Dataset.

	Model name	Subnetworks	Merging method	Input types	Accuracy ± std (%)
Single models	1D CNN	N/A	N/A	raw input	85.75 ± 0.75
	2D CNN	N/A	N/A	spectrogram	89.77 ± 0.97
	LSTM	N/A	N/A	raw input	78.72 ± 1.43
Hybrid models	standard 1D CNN-LSTM	1D CNN LSTM	series	raw input	84.28 ± 1.76
	Ordóñez-Roggen 1D CNN-LSTM	1D CNN LSTM	series	raw input	82.06 ± 0.90
	alternative 1D CNN-LSTM	1D CNN LSTM	parallel	raw input + raw input	84.60 ± 0.94
	proposed 2D CNN-LSTM	2D CNN LSTM	parallel	spectrogram + raw input	92.95 ± 0.47

probability enables different neurons to extract features independently and allows the development of a more robust and generalizable model.

The confusion matrix of the UCI HAR Dataset in Fig. 14 indicates that the model has similar recognition accuracies for all six activities (Recall that this dataset is somewhat imbalanced with comparable but unequal number of samples from each activity type.). On the other hand, the confusion matrix for the DSA Dataset in Fig. 17 shows that the activity 18 (jumping) has a lower recognition rate compared to the other activities. It is incorrectly classified as activity 6 (descending stairs) in 39 out of 60 instances.

4.4. Ablation studies

We have conducted ablation studies to show the contribution of the proposed method on the classification performance. We have considered three key hyper-parameters in these experiments which underlie the innovation that our proposed model brings. These hyper-parameters are the connection type of the branches (series or parallel), type of input (raw data or their spectrograms), and the type of branch (LSTM or 2D CNN). In our experiments, we have modified these hyper-parameters to see their effect on the classification performance of the proposed model. Tables 8 and 9 display the results of the ablation studies for the two datasets.

4.4.1. Connection type of branches

To investigate the effect of the connection type, we first conducted experiments using a series connection of 2D CNN and LSTM modules where the output of the 2D CNN is given as input to the LSTM. Since such an architecture can receive only one type of input, we have given either the raw signals or the spectrogram as input. In the former case, all four performance metrics are below 90% for the UCI-HAR Dataset. When we use spectrogram images as input, resulting performance improves and is similar to that of the parallel model with two spectrogram inputs. This validates other experiments which usually display better results with the spectrogram input type. We also repeated the experiments by changing the order of the 2D CNN and LSTM modules which lowered the performance metrics in general. The main reason could be that such an architecture does not have the benefit of having a deep learning 2D CNN module at the input to receive and process the signals.

4.4.2. Input type

We have already considered feeding the two input types to a series connection in the previous part. Here, we consider network architectures with two branches in parallel which can receive different input types. Our proposed model receives two types of input where the raw sensor signals and their spectrograms are given as input to the LSTM and 2D CNN branches, respectively. To investigate the effect of the input type, we have conducted experiments with four possible input combinations to a parallel structure, namely, raw–raw, raw–spectrogram, spectrogram–raw, and spectrogram–spectrogram.

When we use raw signals as input to both branches, classification performance is similar to that of a single-branch model comprising the branch with the higher classification accuracy. When we use spectrogram images as input to both the LSTM and 2D CNN branches, classification accuracy decreases to a level similar to that of a single

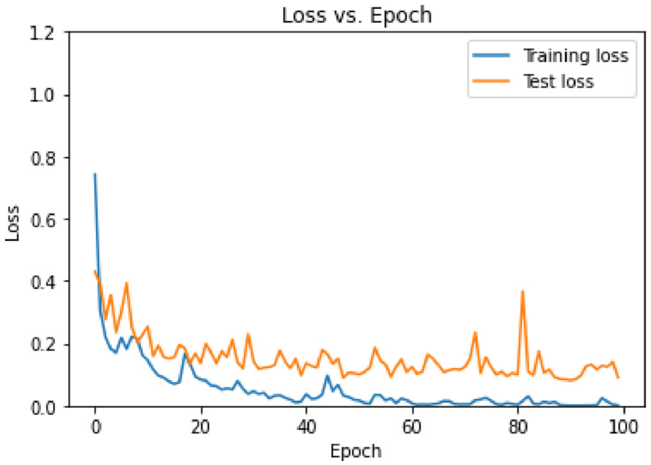


Fig. 12. Loss versus epoch graph of the proposed model based on the test set of the UCI HAR Dataset.

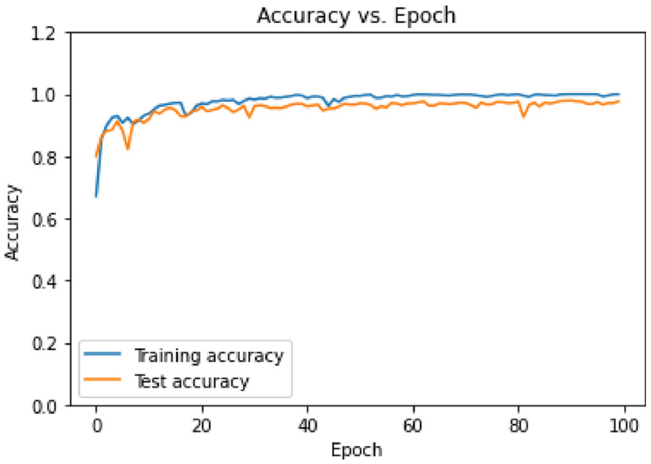


Fig. 13. Accuracy versus epoch graph of the proposed model based on the test set of the UCI HAR Dataset.

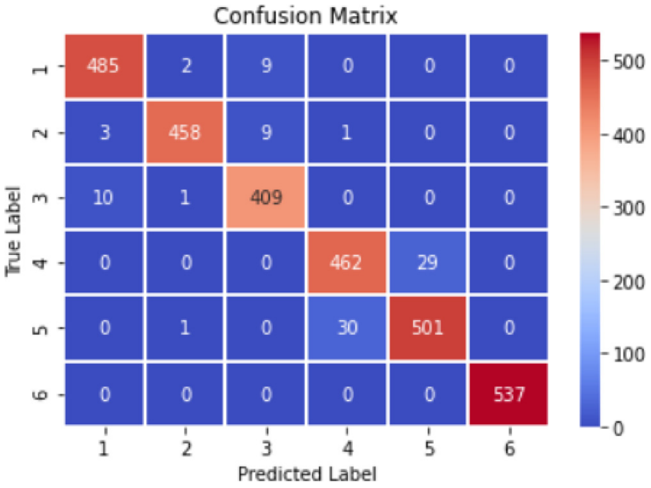


Fig. 14. Confusion matrix of the proposed model based on the test set of the UCI HAR Dataset.

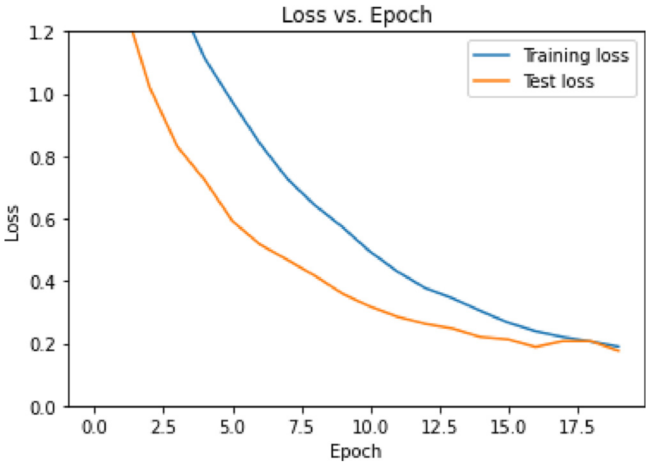


Fig. 15. Loss versus epoch graph of the proposed model based on testing with the second subject's data of the DSA Dataset.

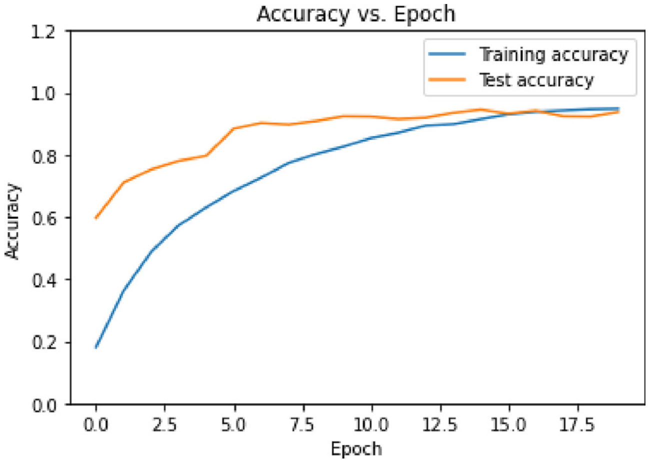


Fig. 16. Accuracy versus epoch graph of the proposed model based on testing with the second subject's data of the DSA Dataset.

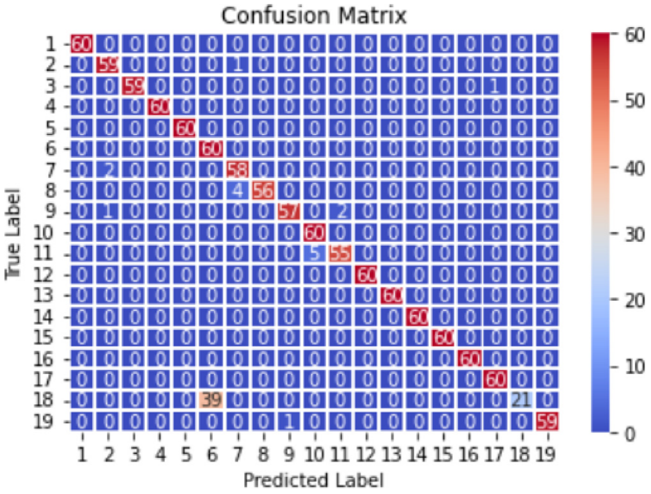


Fig. 17. Confusion matrix of the proposed model based on testing with the second subject's data of the DSA Dataset.

2D CNN model with spectrogram input. This means that the LSTM branch does not extract additional features from the spectrogram that complement those of the 2D CNN branch. Therefore, diversity in feature extraction is low and classification performance is similar to that of

Table 8
Ablation study results for the proposed model on the UCI HAR Dataset.

Removed branch(es)	Remaining branch(es)	Merging method	Input types	Accuracy ± std (%)	F_1 Score ± std (%)	Recall ± std (%)	Precision ± std (%)
none	2D CNN, LSTM	series	raw	89.19 ± 1.45	89.46 ± 1.43	89.47 ± 1.45	89.68 ± 1.38
none	2D CNN, LSTM	series	spectrogram	94.35 ± 0.47	94.36 ± 0.49	94.33 ± 0.51	94.57 ± 0.33
none	2D CNN, LSTM	series (reverse)	raw	85.28 ± 2.61	85.99 ± 2.61	85.65 ± 2.58	86.83 ± 2.27
none	2D CNN, LSTM	series (reverse)	spectrogram	85.06 ± 4.18	85.64 ± 4.29	85.43 ± 4.11	86.35 ± 4.34
none	2D CNN, LSTM	parallel	raw + raw	89.70 ± 0.80	90.04 ± 0.81	90.03 ± 0.77	90.43 ± 0.75
none	2D CNN, LSTM	parallel	raw + spectrogram	90.17 ± 0.48	90.40 ± 0.49	90.39 ± 0.51	90.58 ± 0.48
none	2D CNN, LSTM	parallel	spectrogram + raw	95.66 ± 0.63	95.62 ± 0.65	95.67 ± 0.63	95.65 ± 0.63
none	2D CNN, LSTM	parallel	spectrogram + spectrogram	93.91 ± 0.73	93.92 ± 0.73	93.94 ± 0.72	94.08 ± 0.69
2D CNN	LSTM	N/A	raw	88.63 ± 0.95	89.11 ± 0.87	88.92 ± 0.95	89.62 ± 0.68
2D CNN	LSTM	N/A	spectrogram	85.03 ± 4.43	83.53 ± 5.80	85.05 ± 4.58	87.99 ± 2.97
LSTM	2D CNN	N/A	raw	89.99 ± 0.48	90.18 ± 0.49	90.18 ± 0.53	90.35 ± 0.41
LSTM	2D CNN	N/A	spectrogram	93.65 ± 0.92	93.66 ± 0.93	93.66 ± 0.88	93.90 ± 0.95
2D CNN, LSTM	none	N/A	spectrogram + raw	81.67 ± 0.30	81.30 ± 0.32	81.33 ± 0.32	81.71 ± 0.31

Table 9
Ablation study results for the proposed model on the DSA Dataset.

Removed branch(es)	Remaining branch(es)	Merging method	Input types	Accuracy ± std (%)
none	2D CNN, LSTM	series	raw	56.86 ± 1.36
none	2D CNN, LSTM	series	spectrogram	89.46 ± 0.74
none	2D CNN, LSTM	series (reverse)	raw	67.54 ± 2.00
none	2D CNN, LSTM	series (reverse)	spectrogram	45.00 ± 6.61
none	2D CNN, LSTM	parallel	raw + raw	69.90 ± 0.96
none	2D CNN, LSTM	parallel	raw + spectrogram	68.85 ± 1.76
none	2D CNN, LSTM	parallel	spectrogram + raw	92.95 ± 0.47
none	2D CNN, LSTM	parallel	spectrogram + spectrogram	90.91 ± 0.47
2D CNN	LSTM	N/A	raw	72.91 ± 1.88
2D CNN	LSTM	N/A	spectrogram	63.45 ± 2.88
LSTM	2D CNN	N/A	raw	60.06 ± 0.49
LSTM	2D CNN	N/A	spectrogram	90.67 ± 0.70
2D CNN, LSTM	none	N/A	spectrogram + raw	89.89 ± 0.47

a single 2D CNN model. Thus, we can state that using the same input type for both branches does not bring additional improvement in performance compared to using a single model which can receive only one type of input.

Lastly, we have swapped the input types of the two branches so that we used spectrograms as input to the LSTM and the raw signals as input to the 2D CNN branch. This input combination and the raw–raw input combination gave worse classification accuracy compared to the other two input combinations. Overall, our proposed model results in the best classification performance. Therefore, we conclude that using spectrogram input for the 2D CNN branch and raw input for the LSTM branch is the most suitable way of feeding inputs to the proposed architecture.

4.4.3. Removal of branches

In this part, we investigate the effect of the removing one or both of the branches of the proposed parallel structure. Initially, we remove the LSTM branch and test the classification performance of the remaining model. When we remove either the LSTM or the 2D CNN branch, we observe performance degradation. This indicates that using two branches with the proposed architecture increases the classification performance of the model. Also, we observe that the effect of removing the 2D CNN branch is greater than that of removing the LSTM branch. We also repeated these experiments by reversing the input types, which affected the performance negatively. Lastly, we conducted an experiment by removing both the LSTM and 2D CNN branches. We combined the flattened version of the spectrogram and the raw signals and fed this to a fully connected neural network. For the UCI HAR Dataset, this model gives the lowest accuracy as expected. For the DSA Dataset, it ranks the fourth, following the proposed model, parallel structure with spectrograms at both inputs, and the 2D CNN with spectrogram input. Surprisingly, it gives better results compared to the two single LSTM models and the 2D CNN with raw input. When we consider that the DSA Dataset was acquired from a smaller number of participants (eight subjects) performing a larger number of activities compared to the UCI-HAR Dataset, it is expected that the deep models might have some difficulties learning from a limited number of samples per activity. However, our proposed model overcomes this difficulty and achieves learning from a limited amount of samples as well, providing the highest classification accuracy for both datasets.

4.5. Comparison of computational cost and complexity

Given that the typical wearable device has limited computational resources, memory and computational requirements of DL models should be considered carefully for wearable device applications. In Tables 10 and 11, we display the total number of parameters, model size, total training time, testing time per sample, and the total number of floating point operations (FLOPs) per sample for the seven models that we have implemented and compared in this study. The total number of parameters is largest for the 1D CNN model for both datasets. The alternative 1D CNN-LSTM model also contains a large number of parameters. The required memory space (model size) to deploy these models varies between 2.37–61.88 MB. Among the models implemented to process the UCI HAR Dataset, the proposed model is the third smallest one after 2D CNN and Ordóñez and Roggen's 1D CNN-LSTM in terms of the total number of parameters and the model size. It is slower to train compared to single 1D CNN and 2D CNN models and among the four hybrid models, it ranks the third, following Ordóñez and Roggen's 1D CNN-LSTM and the standard 1D CNN-LSTM which take less time to train. The model size and the training time of the proposed model in processing the DSA Dataset are not as favorable as those of the proposed model developed for the UCI HAR Dataset. However, we can say that it still has similar complexity measures compared to the

Table 10
Complexities of the models implemented to process the UCI HAR Dataset.

	Model name	Total number of parameters	Model size (MB)	Total training Time (s)	Testing time per sample (ms)	FLOPs (million)
Single models	1D CNN	5,402,340	61.88	353	0.81	2.70
	2D CNN	302,976	3.51	75	0.10	0.15
	LSTM	2,498,327	28.65	2,055	18.98	2.77
Hybrid models	standard 1D CNN-LSTM	1,630,916	18.72	242	0.82	1.84
	Ordóñez-Roggen 1D CNN-LSTM	525,126	4.07	142	0.68	0.39
	alternative 1D CNN-LSTM	3,733,560	42.84	1,390	5.63	2.26
	proposed 2D CNN-LSTM	767,110	8.88	1,236	1.60	0.43

Table 11
Complexities of the models implemented to process the DSA Dataset.

	Model name	Total number of parameters	Model size (MB)	Total training Time (s)	Testing time per sample (ms)	FLOPs (million)
Single models	1D CNN	5,404,163	61.73	25	0.28	2.71
	2D CNN	227,293	2.60	14	0.16	0.11
	LSTM	202,929	2.37	79	1.04	0.20
Hybrid models	standard 1D CNN-LSTM	754,419	8.69	28	0.45	0.42
	Ordóñez-Roggen 1D CNN-LSTM	1,095,379	8.42	41	1.30	0.68
	alternative 1D CNN-LSTM	1,239,742	14.12	88	0.70	0.64
	proposed 2D CNN-LSTM	1,794,638	20.50	118	2.90	1.25

Table 12
Comparison of accuracy performance metrics of the proposed model and the related work for processing the UCI HAR Dataset.

Reference study	Model	Accuracy (%)
Present paper	2D CNN-LSTM	95.66 ± 0.63
Yen et al. (2020)	1D CNN	95.99
Xia et al. (2020)	LSTM-2D CNN	95.78
Tufek et al. (2020)	LSTM	93.70
Mutegeki and Han (2020)	1D CNN-LSTM	92.13
Deep and Zheng (2019)	1D CNN-LSTM	93.40

other six models. When we consider the performance improvement the proposed model brings, this difference in complexity can be tolerated. In case it may not be tolerable for on-device computing in a particular application, model compression or computation off-loading methods can be employed (Koşar, 2022).

The testing times of the models vary between 0.10–18.98 ms per sample on a regular user laptop CPU [Intel(R) Core(TM) i7-10510U CPU]. Hence, the models can be considered sufficiently fast to be used in real-time applications. The total number of FLOPs per sample changes between 0.11–2.77 million.

4.6. Comparison of our results with existing studies

We review the results of five recent studies (Yen et al., 2020; Xia et al., 2020; Tufek et al., 2020; Mutegeki and Han, 2020; Deep and Zheng, 2019) that use the UCI HAR Dataset (Table 12) and four recent works (Yurtman and Barshan, 2017; Yurtman et al., 2018; Barshan and Yurtman, 2020; Yurtman et al., 2021) our research group has conducted that process the DSA Dataset (Table 13). However, since there are some differences among these studies in terms of the implementation and the performance evaluation details, the results of these studies are not totally comparable with those of the proposed model.

Mutegeki and Han (2020) and Deep and Zheng (2019) both show that the standard 1D CNN-LSTM model improves the performance of the single LSTM model. Our results also confirm this: When the standard 1D CNN-LSTM model is used with the UCI HAR Dataset, there is 0.85–1.18% improvement in the performance metrics whereas with the DSA Dataset, the accuracy improves by 5.56% compared to the LSTM. However, these two studies do not make a performance comparison with the 1D CNN. Ordóñez and Roggen (2016) demonstrate that the standard 1D CNN-LSTM improves the performance of 1D CNN. Consistent with this, our results in Table 6 indicate that there is improvement by 0.14–0.28% in the performance metrics for the UCI HAR Dataset. However, in Table 7, we observe that the accuracy metric for the standard 1D CNN-LSTM is lower than that of 1D CNN by 1.47%, with larger standard deviation. This could be due to some differences in the implemented models: The standard 1D CNN-LSTM that we developed for processing the UCI HAR Dataset contains three layers (two layers of 1D CNN followed by a LSTM layer) whereas the 1D CNN comprises four layers. When we make the comparison based on the DSA Dataset, we simply add a LSTM layer after the 1D CNN which is the only difference between the 1D CNN and the 1D CNN-LSTM models for this dataset. Hence, in the UCI HAR Dataset, we compare the standard 1D CNN-LSTM with a deeper 1D CNN with four layers, whereas in the DSA Dataset, we compare the sequential 1D CNN and LSTM layers that form the standard 1D CNN-LSTM with a single 1D CNN layer. On the other hand, Ordóñez and Roggen (2016) replace the dense layers of the 1D CNN with LSTM layers. In other words, Ordóñez and Roggen (2016) compare the effect of using LSTM or dense layers after four 1D CNN layers whereas the present paper compares the effect of using LSTM or 1D CNN layers after two 1D CNN layers in processing the UCI HAR Dataset. For the DSA Dataset, we compare the effect of including or not including a LSTM layer after a 1D CNN layer.

Some other differences in the implementation for the studies that use the UCI HAR Dataset are as follows: In creating segmented data,

Table 13
Comparison of accuracy performance metrics of the proposed model and the related work for processing the DSA Dataset.

Reference study	Model	Accuracy ± std (%)
Present paper	2D CNN-LSTM	92.95 ± 0.47
Yurtman and Barshan (2017)	SVM	90.36 ± 2.28
	ANN	89.67 ± 3.63
Yurtman et al. (2018)	ANN	90.93 ± 3.95 (all activities) 87.45 ± 11.69 (stationary) 91.86 ± 3.85 (non-stationary)
Barshan and Yurtman (2020), Yurtman et al. (2021)	ANN	90.93 ± 3.95 (result used from Yurtman et al. (2018))

Yen et al. (2020) use a window size of 256 while the other studies listed in the table, including ours, use a 128-length sliding window. The number of test samples provided in Xia et al. (2020) is not the same as in the other studies which indicates that the way the participants' data are split into training and test sets is different than in the other studies. Moreover, in some earlier studies, it is not clear which set is used for hyper-parameter tuning. Some studies may allocate a separate validation set for hyper-parameter tuning while others may not. In this case, the former group would be disadvantaged compared to the latter. Lastly, since DL models are stochastic in nature, we typically get a different result at each repetition of an experiment. For a fair comparison, providing the average and the standard deviation values of the performance metrics would be more appropriate.

In Table 13, we compare the accuracy result obtained using the DSA Dataset with the results of our previous studies that use the same dataset but implement ML algorithms instead of DL models. In the table, we have chosen to present the results of the most relevant ML classifier used which is artificial neural networks (ANNs). Among the seven state-of-the-art ML classifiers implemented in Yurtman et al. (2018), ANNs result in the highest accuracy of 90.93%, followed by Support Vector Machines (SVMs) with 90.80%. On the other hand, among the four ML classifiers compared in Yurtman and Barshan (2017), SVMs are slightly superior to ANNs with respective accuracy figures of 90.36% and 89.67%. Yurtman and Barshan (2017), besides the DSA Dataset, tests the ML algorithms on four other publicly available datasets for which the accuracy varies between 42.35% and 91.03%. Yurtman et al. (2018) divide the 19 activities into two broad categories as stationary and non-stationary, for which separate accuracy figures are provided, as well as the accuracy result of 90.93% over all activities for ANN. The works Barshan and Yurtman (2020) and Yurtman et al. (2021) both use the result reported in Yurtman et al. (2018) as reference for comparison with the performances of the position and orientation invariant algorithms developed in those studies. According to this table, using DL models instead of ML classifiers can improve the classification accuracy by up to 3.28%. We also note that the standard deviation of the accuracy of the proposed model is considerably lower than those of the previous studies presented in this table.

Although there is not much common basis to make a fair comparison between the studies reviewed above in two groups, in any case, we have presented the results of recent studies together with our results in Tables 12 and 13. Once again, it should be noted that for the reasons explained above, this comparison is not completely like-for-like.

5. Conclusions

We have proposed a novel 2D CNN-LSTM hybrid architecture that enables extracting a broader range of representative features, resulting in higher activity recognition accuracy. Using spectrograms as input for the 2D CNN branch of the new model resulted in considerable performance improvement. We compared the performance of the proposed model with six existing DL models: 1D CNN, 2D CNN, LSTM, standard 1D CNN-LSTM, 1D CNN-LSTM model proposed in Ordóñez and Roggen (2016), and the alternative 1D CNN-LSTM. All seven models were implemented to process two publicly available datasets (UCI HAR and DSA Datasets). The proposed model proved to be superior in terms of the accuracy, F_1 score, recall, and precision performance metrics, while offering an acceptable level of complexity. If complexity is a limiting issue in a specific application, 2D CNN model would be the second best choice, with lower complexity at the expense of some accuracy degradation. Our current work is focused on minimizing the computational complexity of the proposed model to enable real-time on-device computation for resource-limited wearables without a significant degradation in the activity recognition performance (Koşar, 2022). In future work, attention layers and transformer architectures for activity recognition can be investigated.

CRediT authorship contribution statement

Enes Koşar: Conceptualization, Methodology, Software, Investigation, Validation, Writing – original draft. **Billur Barshan:** Conceptualization, Methodology, Supervision, Resources, 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

We have used two publicly available datasets in our work, one of which was acquired by our research group. These two datasets are cited by the references Altun and Barshan (2013, 2019), and Reyes-Ortiz et al. (2015) within which the links for access to the datasets are provided as well.

Appendix

Optimum hyper-parameter values for five of the existing models that we have implemented for comparison with the proposed 2D CNN-LSTM model are given in Tables A.1–A.5 for processing the UCI HAR Dataset, and in Tables A.6–A.10 for using the DSA Dataset. For the 1D CNN-LSTM model proposed by Ordóñez and Roggen (2016), we have used the hyper-parameter values provided in the reference; we did not conduct hyper-parameter tuning.

Table A.1
Optimized hyper-parameters of the 1D CNN in processing the UCI HAR Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer 1 filter number	255	[4, 256]
CNN layer 2 filter number	251	[4, 256]
CNN layer 3 filter number	62	[4, 256]
CNN layer 4 filter number	155	[4, 256]
CNN layer 1 filter width	4	[3, 7]
CNN layer 2 filter width	7	[3, 7]
CNN layer 3 filter width	6	[3, 7]
CNN layer 4 filter width	5	[3, 7]
dense layer neuron number	242	[8, 512]
dropout layer 1 probability	0.33500743	[0.1, 0.9]
dropout layer 2 probability	0.21856031	[0.1, 0.9]
dropout layer 3 probability	0.12632442	[0.1, 0.9]
dropout layer 4 probability	0.70007760	[0.1, 0.9]
learning rate	4.40541658×10^{-5}	[10^{-6}, 0.1]

Table A.2
Optimized hyper-parameters of the 2D CNN in processing the UCI HAR Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer 1 filter number	7	[4, 256]
CNN layer 2 filter number	23	[4, 256]
CNN layer 1 filter width	4	[3, 7]
CNN layer 2 filter width	7	[3, 7]
dense layer 1 neuron number	72	[8, 512]
dense layer 2 neuron number	493	[8, 512]
dropout layer 1 probability	0.11793672	[0.1, 0.9]
dropout layer 2 probability	0.14386716	[0.1, 0.9]
dropout layer 3 probability	0.79227985	[0.1, 0.9]
learning rate	0.00354435	[10^{-6}, 0.1]

Table A.3
Optimized hyper-parameters of the LSTM in processing the UCI HAR Dataset.

Hyper-parameter	Optimum value	Search interval
LSTM layer 1 neuron number	474	[8, 512]
LSTM layer 2 neuron number	362	[8, 512]
LSTM layer 3 neuron number	156	[8, 512]
dense layer neuron number	275	[8, 512]
dropout layer 1 probability	0.13905174	[0.1, 0.9]
dropout layer 2 probability	0.28673553	[0.1, 0.9]
dropout layer 3 probability	0.51215028	[0.1, 0.9]
learning rate	0.00011890	[10^{-6}, 0.1]

Table A.4
Optimized hyper-parameters of the standard 1D CNN-LSTM in processing the UCI HAR Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer 1 filter number	180	[4, 256]
CNN layer 2 filter number	25	[4, 256]
CNN layer 1 filter width	6	[3, 7]
CNN layer 2 filter width	7	[3, 7]
LSTM layer neuron number	506	[8, 512]
dense layer neuron number	309	[8, 512]
dropout layer 1 probability	0.25003751	[0.1, 0.9]
dropout layer 2 probability	0.40697846	[0.1, 0.9]
dropout layer 3 probability	0.48942447	[0.1, 0.9]
dropout layer 4 probability	0.44661285	[0.1, 0.9]
learning rate	0.00015457	[10^{-6}, 0.1]

Table A.5
Optimized hyper-parameters of the alternative 1D CNN-LSTM in processing the UCI HAR Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer 1 filter number	154	[4, 256]
CNN layer 2 filter number	161	[4, 256]
CNN layer 3 filter number	96	[4, 256]
CNN layer 4 filter number	92	[4, 256]
CNN layer 1 filter width	7	[3, 7]
CNN layer 2 filter width	6	[3, 7]
CNN layer 3 filter width	4	[3, 7]
CNN layer 4 filter width	6	[3, 7]
LSTM layer 1 neuron number	63	[8, 512]
LSTM layer 2 neuron number	291	[8, 512]
LSTM layer 3 neuron number	101	[8, 512]
dense layer 1 neuron number	493	[8, 512]
dense layer 2 neuron number	239	[8, 512]
dropout layer 1 probability	0.23502093	[0.1, 0.9]
dropout layer 2 probability	0.25219255	[0.1, 0.9]
dropout layer 3 probability	0.13254391	[0.1, 0.9]
dropout layer 4 probability	0.18250190	[0.1, 0.9]
dropout layer 5 probability	0.35352521	[0.1, 0.9]
dropout layer 6 probability	0.17141114	[0.1, 0.9]
dropout layer 7 probability	0.42956279	[0.1, 0.9]
learning rate	0.00033614	[10^{-6}, 0.1]

Table A.6
Optimized hyper-parameters of the 1D CNN in processing the DSA Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer filter number	83	[4, 256]
CNN layer filter width	5	[3, 7]
dense layer 1 neuron number	498	[8, 512]
dense layer 2 neuron number	341	[8, 512]
dropout layer probability	0.86982345	[0.1, 0.9]
learning rate	0.00172552	[10^{-6}, 0.1]

Table A.7
Optimized hyper-parameters of the 2D CNN in processing the DSA Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer filter number	27	[4, 256]
CNN layer filter width	3	[3, 7]
dense layer 1 neuron number	66	[8, 512]
dense layer 2 neuron number	141	[8, 512]
dropout layer probability	0.26342746	[0.1, 0.9]
learning rate	0.00026890	[10^{-6}, 0.1]

Table A.8
Optimized hyper-parameters of the LSTM in processing the DSA Dataset.

Hyper-parameter	Optimum value	Search interval
LSTM layer neuron number	157	[8, 512]
dense layer 1 neuron number	122	[8, 512]
dense layer 2 neuron number	391	[8, 512]
dropout layer probability	0.17112627	[0.1, 0.9]
learning rate	0.00306971	[10^{-6}, 0.1]

Table A.9
Optimized hyper-parameters of the standard 1D CNN-LSTM in processing the DSA Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer filter number	66	[4, 256]
CNN layer filter width	5	[3, 7]
LSTM layer neuron number	104	[8, 512]
dense layer 1 neuron number	68	[8, 512]
dense layer 2 neuron number	21	[8, 512]
dropout layer probability	0.10679134	[0.1, 0.9]
learning rate	0.00212826	[10^{-6}, 0.1]

Table A.10
Optimized hyper-parameters of the alternative 1D CNN-LSTM in processing the DSA Dataset.

Hyper-parameter	Optimum value	Search interval
CNN layer filter number	76	[4, 256]
CNN layer filter width	7	[3, 7]
LSTM layer neuron number	58	[8, 512]
dense layer 1 neuron number	127	[8, 512]
dense layer 2 neuron number	110	[8, 512]
dense layer 3 neuron number	392	[8, 512]
dropout layer probability	0.55504952	[0.1, 0.9]
learning rate	0.00074461	[10^{-6}, 0.1]

References

Altun, K., Barshan, B., 2013. Daily and sports activities dataset. In: UCI Machine Learning Repository. School Inf. Comput. Sci. Univ. California at Irvine, Irvine, CA, U.S.A., available online: http://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities.

Altun, K., Barshan, B., 2019. Daily and sports activities dataset. IEEE Data Port http://dx.doi.org/10.21227/at1v-6f84.

Anguita, D., Ghio, A., Oneto, L., Parra Perez, X., Reyes-Ortiz, J.L., 2013. A public domain dataset for human activity recognition using smartphones. In: Proc. 21st European Symp. Artificial Neural Networks, Computational Intelligence and Machine Learning. ESANN, Bruges, Belgium, 24–26 April 2013, ISBN: 978-2-87419-081-0, pp. 437–442.

Barshan, B., Yurtman, A., 2020. Classifying daily and sports activities invariantly to the positioning of wearable motion sensor units. IEEE Internet Things J. 7 (6), 4801–4815. http://dx.doi.org/10.1109/jiot.2020.2969840.

Barut, O., Zhou, L., Luo, Y., 2020. Multitask LSTM model for human activity recognition and intensity estimation using wearable sensor data. IEEE Internet Things J. 7 (9), 8760–8768. http://dx.doi.org/10.1109/jiot.2020.2996578.

Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., Liu, Y., 2022. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. ACM Comput. Surv. 54 (4), 77. http://dx.doi.org/10.1145/3447744.

Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder–decoder approaches. arXiv:1409.1259v2.

Chung, S., Lim, J., Noh, K.J., Kim, G., Jeong, H., 2019. Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning. Sensors (MDPI) 19 (7), 1716. http://dx.doi.org/10.3390/s19071716.

Deep, S., Zheng, X., 2019. Hybrid model featuring CNN and LSTM architecture for human activity recognition on smartphone sensor data. In: Proc. 20th Int. Conf. Parallel and Distributed Computing, Applications and Technologies. PDCAT, 5–7 December 2019, Gold Coast, QLD, Australia, pp. 262–267. http://dx.doi.org/10.1109/pdcat46702.2019.00055.

Dhiman, C., Vishwakarma, D.K., 2019. A review of state-of-the-art techniques for abnormal human activity recognition. Eng. Appl. Artif. Intell. 77, 21–45. http://dx.doi.org/10.1016/j.engappai.2018.08.014.

Gil-Martín, M., San-Segundo, R., Fernández-Martínez, F., Ferreiros-López, J., 2020. Improving physical activity recognition using a new deep learning architecture and post-processing techniques. Eng. Appl. Artif. Intell. 92, 103679. http://dx.doi.org/10.1016/j.engappai.2020.103679.

Haktanır, E., Kahraman, C., 2022. A novel picture fuzzy CRITIC & REGIME methodology: Wearable health technology application. Eng. Appl. Artif. Intell. 113, 104942. http://dx.doi.org/10.1016/j.engappai.2022.104942.

Hamad, R.A., Yang, L., Woo, W.L., Wei, B., 2020. Joint learning of temporal models to handle imbalanced data for human activity recognition. Appl. Sci. 10 (15), 5293. http://dx.doi.org/10.3390/app10155293.

Han, C., Zhang, L., Tang, Y., Huang, W., Min, F., He, J., 2022. Human activity recognition using wearable sensors by heterogeneous convolutional neural networks. Expert Syst. Appl. 198, 116764. http://dx.doi.org/10.1016/j.eswa.2022.116764.

Huang, W., Zhang, L., Wang, S., Wu, H., Song, A., 2022a. Deep ensemble learning for human activity recognition using wearable sensors via filter activation. ACM Trans. Embedded Comput. Syst. 22 (1), 15. http://dx.doi.org/10.1145/3551486.

Huang, W., Zhang, L., Wu, H., Min, F., Song, A., 2022b. Channel-equalization-HAR: A light-weight convolutional neural network for wearable sensor based human activity recognition. IEEE Trans. Mob. Comput. http://dx.doi.org/10.1109/tmc.2022.3174816.

Huynh-The, T., Hua, C.-H., Tu, N.A., Kim, D.-S., 2021. Physical activity recognition with statistical-deep fusion model using multiple sensory data for smart health. IEEE Internet Things J. 8 (3), 1533–1543. http://dx.doi.org/10.1109/jiot.2020.3013272.

Ito, C., Cao, X., Shuzo, M., Maeda, E., 2018. Application of CNN for human activity recognition with FFT spectrogram of acceleration and gyro sensors. In: Proc. ACM Int. Joint Conf. and Int. Symp. Pervasive and Ubiquitous Computing and Wearable Computers. UBICOMP, 8–12 October 2018, Singapore, Singapore, pp. 1503–1510. http://dx.doi.org/10.1145/3267305.3267517.

Kim, Y., Moon, T., 2016. Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks. IEEE Geosci. Remote Sens. Lett. 13 (1), 8–12. http://dx.doi.org/10.1109/lgrs.2015.2491329.

Kim, Y., Toomajian, B., 2016. Hand gesture recognition using micro-Doppler signatures with convolutional neural network. IEEE Access 4, 7125–7130. http://dx.doi.org/10.1109/access.2016.2617282.

Koşar, E., 2022. A Memory Efficient Novel Deep Learning Architecture Enabling Diverse Feature Extraction on Wearable Motion Sensor Data (M.S. thesis). Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara, Turkey, available online: http://hdl.handle.net/11693/110573.

Lattanzi, E., Freschi, V., 2020. Evaluation of human standing balance using wearable inertial sensors: A machine learning approach. Eng. Appl. Artif. Intell. 94, 103812. http://dx.doi.org/10.1016/j.engappai.2020.103812.

Lawal, I.A., Bano, S., 2020. Deep human activity recognition with localisation of wearable sensors. IEEE Access 8, 155060–155070. http://dx.doi.org/10.1109/access.2020.3017681.

Li, Y., Zhang, S., Zhu, B., Wang, W., 2020. Accurate human activity recognition with multi–task learning. CCF Trans. Pervas. Comput. Interact. 2, 288–298. http://dx.doi.org/10.1007/s42486-020-00042-2.

Lv, T., Wang, X., Jin, L., Xiao, Y., Song, M., 2020. Margin-based deep learning networks for human activity recognition. Sensors (MDPI) 20 (7), 1871. http://dx.doi.org/10.3390/s20071871.

Mekruksavanich, S., Jitpattanakul, A., 2020. Smartwatch-based human activity recognition using hybrid LSTM network. In: Proc. IEEE Sensors Conf., 25–28 October 2020. Rotterdam, Netherlands, http://dx.doi.org/10.1109/sensors47125.2020.9278630.

Mekruksavanich, S., Jitpattanakul, A., 2021. LSTM networks using smartphone data for sensor-based human activity recognition in smart homes. Sensors (MDPI) 21 (5), 1636. http://dx.doi.org/10.3390/s21051636.

Mukherjee, D., Mondal, R., Singh, P.K., Sarkar, R., Bhattacharjee, D., 2020. EnsemConvNet: A deep learning approach for human activity recognition using smartphone sensors for healthcare applications. Multimedia Tools Appl. 79 (41), 31663–31690. http://dx.doi.org/10.1007/s11042-020-09537-7.

Mutegeki, R., Han, D.S., 2020. A CNN-LSTM approach to human activity recognition. In: Proc. 2nd IEEE Int. Conf. Artificial Intelligence in Information and Communication. ICAIIC, 19–21 February 2020, Fukuoka, Japan, pp. 362–366. http://dx.doi.org/10.1109/icaiic48513.2020.9065078.

Niknejad, N., Ismail, W.B., Mardani, A., Liao, H., Ghani, I., 2020. A comprehensive overview of smart wearables: The state of the art literature, recent advances, and future challenges. Eng. Appl. Artif. Intell. 90, 103529. http://dx.doi.org/10.1016/j.engappai.2020.103529.

Oppenheim, A.V., Schafer, R.W., Buck, J.R., 1999. Discrete-Time Signal Processing, second ed. Prentice Hall, Upper Saddle River, NJ, U.S.A., (see Ch. 10, 714–722 on the time-dependent Fourier transform).

Ordóñez, F.J., Roggen, D., 2016. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors (MDPI) 16 (1), 115. http://dx.doi.org/10.3390/s16010115.

Pardo, L.B., Perez, D.B., Uruñuela, C.O., 2019. Detection of tennis activities with wearable sensors. Sensors (MDPI) 19 (22), 5004. http://dx.doi.org/10.3390/s19225004.

Park, J., Javier, R.J., Moon, T., Kim, Y., 2016. Micro-Doppler based classification of human aquatic activities via transfer learning of convolutional neural networks. Sensors (MDPI) 16 (12), 1990. http://dx.doi.org/10.3390/s16121990.

Peng, L., Chen, L., Ye, Z., Zhang, Y., 2018. AROMA: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors. In: Proc. ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, no. 2. 74. http://dx.doi.org/10.1145/3214277.

Pravallika, R., Sreenivasarao, D., Saheb, S.K., 2020. Deep learning for human activity recognition using on-node sensors. Int. J. Recent Technol. Eng. 8 (5), 607–614. http://dx.doi.org/10.35940/ijrte.E5654.018520.

Qin, Z., Zhang, Y., Meng, S., Qin, Z., Choo, K.-K.R., 2020. Imaging and fusing time series for wearable sensor-based human activity recognition. Inf. Fusion 53, 80–87. http://dx.doi.org/10.1016/j.inffus.2019.06.014.

Ramanujam, E., Perumal, T., Padmavathi, S., 2021. Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review. IEEE Sens. J. 21 (12), 13029–13040. http://dx.doi.org/10.1109/jsen.2021.3069927.

Ravi, D., Wong, C., Lo, B., Yang, G.-Z., 2017. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. IEEE J. Biomed. Health Inf. 21 (1), 56–64. http://dx.doi.org/10.1109/jbhi.2016.2633287.

Reyes-Ortiz, J.L., Anguita, D., Oneto, L., Parra, X., 2015. Smartphone-based recognition of human activities and postural transitions data set. In: UCI Machine Learning Repository. School Inf. Comput. Sci. Univ. California at Irvine, Irvine, CA, U.S.A., available online: http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions.

Reyes-Ortiz, J.L., Oneto, L., Samà, A., Parra, X., Anguita, D., 2016. Transition-aware human activity recognition using smartphones. Neurocomputing 171, 754–767, https://www.sciencedirect.com/science/article/abs/pii/S0925231215010930?via=ihub.

Sena, J., Barreto, J., Caetano, C., Cramer, G., Schwartz, W.R., 2021. Human activity recognition based on smartphone and wearable sensors using multiscale DCNN ensemble. Neurocomputing 444, 226–243. http://dx.doi.org/10.1016/j.neucom.2020.04.151.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical Bayesian optimization of machine learning algorithms. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems: Proc. 25th Int. Conf. Neural Information Processing Syst., vol. 25. NIPS, Lake Tahoe, NV, U.S.A., 3–6 December 2012, Curran Associates Inc., Red Hook, NY, U.S.A..

Tang, Y., Zhang, L., Min, F., He, J., 2023. Multiscale deep feature learning for human activity recognition using wearable sensors. IEEE Trans. Ind. Electron. 70 (2), 2106–2116. http://dx.doi.org/10.1109/tie.2022.3161812.

Tufek, N., Yalcin, M., Altintas, M., Kalaoglu, F., Li, Y., Bahadir, S.K., 2020. Human action recognition using deep learning methods on limited sensory data. IEEE Sens. J. 20 (6), 3101–3112. http://dx.doi.org/10.1109/jsen.2019.2956901.

Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L., 2019. Deep learning for sensor-based activity recognition: A survey. Pattern Recognit. Lett. 119, 3–11. http://dx.doi.org/10.1016/j.patrec.2018.02.010.

Wang, K., He, J., Zhang, L., 2021. Sequential weakly labeled multiactivity localization and recognition on wearable sensors using recurrent attention networks. IEEE Trans. Hum. Mach. Syst. 51 (4), 355–364. http://dx.doi.org/10.1109/thms.2021.3086008.

Wang, H., Zhao, J., Li, J., Tian, L., Tu, P., Cao, T., An, Y., Wang, K., Li, S., 2020. Wearable sensor-based human activity recognition using hybrid deep learning techniques. Secur. Commun. Netw. 2020, 2132138. http://dx.doi.org/10.1155/2020/2132138.

Xia, K., Huang, J., Wang, H., 2020. LSTM-CNN architecture for human activity recognition. IEEE Access 8, 56855–56866. http://dx.doi.org/10.1109/access.2020.2982225.

Yao, S., Hu, S., Zhao, Y., Zhang, A., Abdelzaher, T., 2017. DeepSense: A unified deep learning framework for time-series mobile sensing data processing. In: Proc. 26th Int. Conf. World Wide Web (WWW), 3–7 April 2017, Perth, Australia. pp. 351–360. http://dx.doi.org/10.1145/3038912.3052577.

Yen, C.-T., Liao, J.-X., Huang, Y.-K., 2020. Human daily activity recognition performed using wearable inertial sensors combined with deep learning algorithms. IEEE Access 8, 174105–174114. http://dx.doi.org/10.1109/access.2020.3025938.

Yurtman, A., Barshan, B., 2017. Activity recognition invariant to sensor orientation with wearable motion sensors. Sensors (MDPI) 17 (8), 1838. http://dx.doi.org/10.3390/s17081838.

Yurtman, A., Barshan, B., Fidan, B., 2018. Activity recognition invariant to wearable sensor unit orientation using differential rotational transformations represented by quaternions. Sensors (MDPI) 18 (8), 2725, (Special Issue on Data Analytics and Applications of Wearable Sensors in Healthcare), https://doi.org/10.3390/s18082725.

Yurtman, A., Barshan, B., Redif, S., 2021. Position invariance for wearables: Interchangeability and single-unit usage via machine learning. IEEE Internet Things J. 8 (10), 8328–8342. http://dx.doi.org/10.1109/jiot.2020.3044754.

Zhang, S., Li, Y., Zhang, S., Shahabi, F., Xia, S., Deng, Y., Alshurafa, N., 2022. Deep learning in human activity recognition with wearable sensors: A review on advances. Sensors (MDPI) 22 (4), 1476. http://dx.doi.org/10.3390/s22041476.

Zhu, J., Chen, H., Ye, W., 2020. A hybrid CNN-LSTM network for the classification of human activities based on micro-Doppler radar. IEEE Access 8, 24713–24720. http://dx.doi.org/10.1109/access.2020.2971064.

Zhu, R., Xiao, Z., Li, Y., Yang, M., Tan, Y., Zhou, L., Lin, S., Wen, H., 2019. Efficient human activity recognition solving the confusing activities via deep ensemble learning. IEEE Access 7, 75490–75499. http://dx.doi.org/10.1109/access.2019.2922104.

Enes Koşar received the B.S. degrees in electrical and electronics engineering and in computer engineering from Koç University, Istanbul, Turkey in 2019 and 2020, respectively, and his M.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey in September 2022. Currently, he is working towards the Ph.D. degree in the same department. His current research interests include accurate and computationally efficient neural networks for human activity recognition and deep learning. He received merit-based full scholarship from Koç University (2013) and graduate studies fellowships from TÜBİTAK (2019) and Türk Telekom (2020).

Billur Barshan received the B.S. degrees in electrical engineering and in physics from Boğaziçi University, Istanbul, Turkey, and the M.S., M.Phil., and Ph.D. degrees in electrical engineering from Yale University, New Haven, CT, U.S.A.

After working as a post-doctoral researcher in the Robotics Research Group, University of Oxford, Oxford, U.K., she joined the Faculty of Bilkent University, Ankara Turkey, where she is currently a Professor with the Department of Electrical and Electronics Engineering. Her current research interests include wearable sensing, wearable robots and mechanisms, intelligent sensing, motion capture and analysis, detection and classification of falls, machine/deep learning, pattern classification, and multi-sensor data fusion.

Dr. Barshan received the TÜBİTAK Incentive Award (1998), METU Mustafa Parlar Foundation Research Award (1999), and two best paper awards. She served on the Management Committee of the COST-IC0903 Action MOVE (2010–2013). Currently, she serves on the Editorial Board of *Digital Signal Processing* (Elsevier) journal and as a Guest Editor for *IEEE Selected Topics in Signal Processing*.