I.J. Intelligent Systems and Applications, **2026, 4, 97-115**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2026.04.06

Modeling Human Activity Recognition with Deep Convolutional Neural Networks and Wearable Sensor Data

S. M. Mohidul Islam*
Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh
E-mail: mohid@cse.ku.ac.bd
ORCID iD: https://orcid.org/0000-0002-8705-3636
*Corresponding author

Kamrul Hasan Talukder
Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh
E-mail: khtalukder@ku.ac.bd
ORCID iD: https://orcid.org/0000-0001-5811-4201

Received: January 6, 2026; Revised: March 15, 2026; Accepted: June 21, 2026; Published: August 8, 2026

Abstract: Automatic human activity recognition has many applications in smart environments, including smart homes, smart cities, smart industries, and smart healthcare centers. While performing the activities by the participants, ambient or body-worn sensors can measure physical movements and those data can be used to develop machine learning models for recognizing those activities. In this study, we have proposed a deep convolutional neural network (DCNN) based method for recognizing human activities using body-worn sensors' time-series data after extensive data analysis. High-quality data is generated and balanced using a preprocessing chain for human activity recognition based on data analysis. The preprocessed data is segmented using a constant-size sliding window. We developed several different DCNN models using random searches and based on validation accuracy we selected the best one for further training and testing. The outputs of the selected model serve as the final predicted activities. We assessed our method on three popular and standard datasets: PAMAP2, WISDM_ar_v1.1, and UCI-HAR, and achieved 98.11%, 98.48%, and 93.25% accuracies for subject-dependent case and 90.27%, 94.51%, and 98.67% accuracies for subject-independent case. The performances of the experimental results are measured using several evaluation metrics and measures that demonstrate the strength of the proposed model over the state-of-the-art.

Index Terms: Human Activity Recognition, Time-Series Data, Wearable Sensors, Subject Independence, Leave-One-Person-Out Cross-Validation, Deep Convolutional Neural Networks.

1. Introduction

In our daily lives, we perform many activities such as walking, running, sleeping, cooking, standing, sitting, etc. Humans generally learn these activities in childhood and can perform them without needing anyone's help. These are called activities of daily living (ADL). Some ADLs are performed using an instrument such as cycling, cooking, using a telephone, doing laundry, etc. These are called instrumental activities of daily living (IADL) [1]. Furthermore, among the activities any human performs, some activities involve a single movement or action such as walking, sitting, running, standing, lying, etc. These are known as simple activities. On the other hand, some activities involve multiple movements or actions such as cleaning, washing hands, watering plants, etc. These are known as complex activities [2]. Moreover, among the activities, some activities are static, such as standing, lying, sitting, etc. and some activities are dynamic such as walking, walking upstairs, walking downstairs, jogging, writing, etc. [3].

Recognizing those human activities has many applications in real life such as in smart homes, smart healthcare environments, smart industries, etc. [4]. Consequently, human activity recognition (HAR) has become a significant area for researchers to automatically recognize the activities in smart environments. To identify the activity, most of the existing methods used data from wearable sensors [3, 5-12] while some use data from environmental sensors [7, 13-17]. The wearable sensors that are mostly used are smartphones, smartwatches, smart wristbands, heart rate monitors, Inertial Measurement Units (IMUs, consisting of accelerometer, gyroscope, and magnetometer), etc. [18].

The environmental sensors that are mostly used are cameras, GPS, temperature sensors, light sensors, RFID, etc. [19]. As a machine learning model, many researchers use shallow learning methods [3, 6-9, 13, 20-23] whereas many other researchers use deep learning methods [5, 10-12, 14-16, 24-31] for identifying the activities of humans. In the case of shallow learning-based methods, representative features to be used for machine learning modeling are extracted using domain knowledge. In the case of deep learning-based methods, no labor-intensive feature engineering is required due to the learning capability of this type of algorithm directly from raw input data. HAR methods using classical machine learning model mostly use k-NN [3, 9, 13, 20, 21], Naïve Bayes [3, 7, 9, 20, 21], Decision Tree [3, 9, 20, 22], SVM [3, 9, 20-22], Random forest [3, 6, 22], Multi-layer perceptron [9, 23], Logistic Regression [3, 22], Linear Discriminant Analysis [20, 21], Quadratic Discriminant Analysis [20], Extra Trees [8], Bag tree [21], Hidden Markov Model [7], Joint Boosting [7], Gradient Boosting [22], etc. On the other hand, HAR methods using deep model mostly use CNN [5, 11, 24, 25], LSTM [25], CNN-LSTM [5, 11, 16], CNN-biLSTM [26], ConvLSTM [5, 27], Stacked LSTM [5], Deep Res-Bidir-LSTM [12], BiLSTM [28], LSTM-RNN [24], CNN-GRU [27], Inception GoogLeNet-GRU [10], Deep SRUs-GRUs [29], Attention GRU [30], BiGRU [28], VGG16 [14, 31], VGG19 [14], ResNet-50 [14], Inception-ResNet-v2 [14], Inception-v3 [14], InceptionTime [27], DNN [24], Encoded-Net [11], DCGAN [31], AlexNet [15], etc. We identified several gaps in the state-of-the-art: (1) Though performance is decent in some existing methods, there is still room for improvement (2) Most of the existing works studied either for person-dependent case or for person-independent case but not for both cases (3) Time complexity is not noted in most of the methods (4) Some of the methods work on datasets either with data of only simple activities or with data of less types of activities. Our objectives are to build a HAR system that surpasses the prediction accuracy of state-of-the-art works, to study both person-dependent and person-independent cases, to figure out the speed of execution to identify whether it is suitable for real-time applications, and to conduct experiments on standard publicly available bigger datasets containing both static and dynamic activities, ADLs and IADLs, as well as simple and complex activities whose data are captured by various types of body-worn sensors located in various positions of human body to evaluate the method elaborately.

To attain high precision in recognizing various types of activities, the input sensor data must be clean and for this reason, we elaborately analyzed the data and used enormous pre-processing on the sensor data. In addition, the machine learning model must be proficient in extracting features of various resolutions from input data. For this reason, we proposed a new deep model that can extract such elements successfully using convolutional filters of various sizes. We have proposed a modified DCNN model and this amended architecture is based on the basic structural design of DCNN developed in [32]. Several different DCNN models are produced by varying the hyper-parameters and the best DCNN is selected based on the validation accuracy which in turn offers decent accuracy. Again, to attain a good speed in execution, the set of parameters should be as small as possible in developing the model. For this reason, we proposed such a structural design of the DCNN model where we need to optimize only a small number of hyper-parameters. Moreover, to achieve a user- or person-independent HAR model, the model is evaluated using unseen or new users whose data are not used to train that model; i.e., in the person-independent case, data is divided into train and test subsets based on the user rather than percentage-wise random splitting. Finally, to conduct larger experiments, we have used three publicly available benchmark datasets that contain a large number of raw wearable sensor data and to enormously assess the generalizability and competence of the proposed DCNN method, we have used numerous evaluation measures and metrics for both person-dependent and person-independent cases. The key contributions of this research are as follows:

- We have proposed the modified DCNN model which is grounded on the elementary architecture established in [32]. Using this model, we build a physiological HAR system using wearable sensor data that achieves better prediction accuracy than state-of-the-art methods. To achieve better accuracy, especially in predicting complex human activities, the data is required to be clean, and so, we applied enormous preprocessing-chain on the raw sensor data. As well, the learning approach must be proficient in mining features of various resolutions from raw sensor data. For this, we used several convolution operations using various filters that can mine all of the features successfully. Furthermore, several different DCNN models are produced, and based on the validation accuracy the best one is selected. This assists in discovering the best model that in turn offers better accuracy.
- To attain a high execution speed, we proposed such architecture of the DCNN model which requires only a small number of hyper-parameters to optimize.
- We used several evaluation measures and metrics to comprehensively assess the proficiency and generalizability of the offered approach for both person-dependent and person-independent cases.

The next sections of this paper are structured as follows: Section 2 presents a detailed description of the proposed method. Subsequently, in Section 3, we present the experimental setup and detailed evaluation results. We then outline the discussions and comparisons with the state-of-the-art methods in Section 4. Finally, the conclusion of this paper is drawn in Section 5.

2. Proposed Approach

The research approach used in this HAR work entails key steps as illustrated in Fig. 1. In this proposal, G different deep CNN models are developed. Before disclosing the data to this model, the time-series body-worn motion sensor raw data is preprocessed using a preprocessing chain after enormous data analysis. Then we processed them to extract features by applying a fixed-size sliding window of suitable length. Later, we divided the segmented data into several subsets and selected the best from G distinct DCNN models based on validation accuracy. Finally, the selected DCNN model accepts segmented test data and recognizes the activity type of the participants. The predicted result is assessed using several performance metrics and measures. All of these steps represent supervised classification of human activities which, based on participants' body movements, can contribute to various smart applications such as healthcare applications.

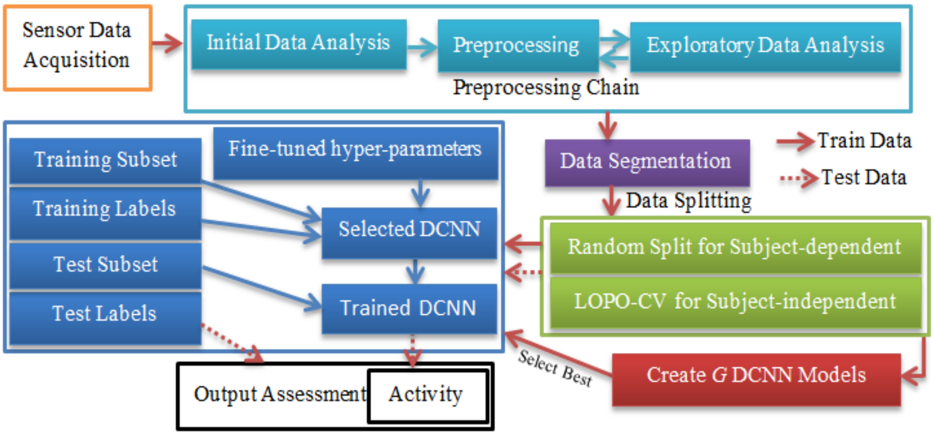


Fig.1. Architecture of the proposed method

2.1. Datasets

We used three publicly available standard human activity datasets to assess our proposed method: PAMAP2 [33], WISDM_ar_v1.1 [34], and UCI-HAR [35]. We selected these benchmark datasets for several reasons: (1) all three datasets contain multivariate, time series, and raw sensor data collected from body-worn sensors rather than ad-hoc sensors placed on the object and/or the environment (2) The volume of data in these datasets is substantial. The performance of machine learning model (especially deep learning model) decreases with small datasets due to overfitting (3) They comprise both simple and complicated real-life human activities (4) The WISDM_ar_v1.1 is a single-sensor dataset whereas the other two are multi-sensor datasets. Multi-sensor setups not only involve multiple sensors but also encompass multiple sensor positions in the participant’s body. Again, the WISDM and UCI-HAR datasets contain data from the Smartphone’s built-in sensors (accelerometer for WISDM, and accelerometer and gyroscope for UCI-HAR) whereas the PAMAP2 dataset contains data from body-worn heart rate monitors and IMUs. Because of using different numbers of sensors, different types of sensors, and different sensor locations on the body; the explored activities in the three datasets are different. The general characteristic of the datasets including their differences is summarized in Table 1.

Table 1. General characteristics of the datasets

Characteristics	**PAMAP2**	**WISDM_ar_v1.1**	**UCI-HAR**
Dataset type	Multi-sensor dataset	Single-sensor dataset	Multi-sensor dataset
Number of subjects	9	36	30
Number of activities	18 (6 of them were optional)	6	6
Sensor sampling rate	100Hz (9 Hz for HR Monitor)	20Hz	50Hz
Sensor used	3 IMUs, 1 Heart rate monitor	accelerometer of the Android smartphone	Accelerometer and gyroscope of the Android smartphone
Sensor placement	On-body: IMUs on the wrist on the dominant arm, on the chest, and on the dominant side's ankle	On-body: smartphone in front trouser pocket (thigh)	On-body: smartphone on the waist
Missing values	Has missing values	has only one missing value	No missing value
Class distribution	imbalanced	imbalanced	Almost balanced
Number of attributes	54	7	11
Number of data	2872533 (raw)	1098204 (raw)	10299 (segmented)

The PAMAP2 dataset contains ADLs, household and sport activities of human (*Rope_jumping, Lying, Sitting, Standing, Walking, Running, Cycling, Nordic_walking, Ascending_stairs, Descending_stairs, Vacuum_cleaning, Ironing, Watching_TV, Computer_work, Car_driving, Folding_laundry, House_cleaning, Playing_soccer*) whereas both the WISDM_ar_v1.1 and UCI-HAR datasets contain ADLs (WISDM_ar_v1.1 (*Walking, Jogging, Upstairs, Downstairs, Sitting, Standing*) and UCI-HAR (*Walking, Walking_upstairs, Walking_downstairs, Sitting, Standing, Laying*)). The above descriptions focus on the real-world applicability of these datasets to the problem at hand.

2.2. Pre-processing Chain

Input data need to be cleaned and balanced enough before learning and prediction. To do this, we analyzed the raw sensor data thoroughly and preprocessed the data accordingly. Initial Data Analysis (IDA) was performed at both univariate and multivariate levels. Careful IDA and accordingly pre-processing the data reduces the risk of misleading or incorrect results. IDA allows the researcher to take a bird's-eye view of the data and tries to make some sense of it [36]. We cleaned and balanced the data using the same preprocessing chain for human activity recognition (PC-HAR) proposed in our previous work [37]. The well-prepared data using the mentioned PC-HAR supports the DCNN model in attaining high prediction accuracy.

A. Cleaning the Data

If any subject performs a small number of the requested activities (and vice versa), we remove the data of that subject (or activity) from the dataset. If any inconsistency is found in the 'timestamps' feature such as tuples with 'timestamp' = 0, then we eliminate those tuples because they do not denote the correct timestamps in sequential data. We exclude the data on transition activities in this study. The irrelevant features (which don't enclose any information that is convenient for the HAR task) are removed before learning and recognition. We eliminate the duplicate instances from the dataset to ensure the validity of the results. If any predictor feature contains missing values in more than 60% of the samples, we eliminate that predictor feature from the dataset because that feature stores a low amount of information for HAR Otherwise, we use imputation to fill in the missing values in that feature. However, if a sample contains any missing data, we remove that sample from the dataset because sufficient data are available.

B. Handling Class Imbalance

If the activity-wise data points are not equally disseminated then there is a necessity for class imbalance handling. In the second column of Tables 2 and 3, we see the class distribution of the raw PAMAP2 and WISDM_ar_v1.1 datasets respectively and we see that there are potential biases toward certain classes in both datasets. So they need to be made balanced. We use the oversampling method, Synthetic Minority Oversampling Technique (SMOTE) for balancing samples in the dataset. In SMOTE, the fabricated samples are created for the minority class by synthesizing the existing samples [38]. This simple approach first randomly selects a minority class sample, *i,* and finds its *k*-nearest neighbors of that class. Then the synthetic sample is generated by randomly selecting one of those *k*-nearest neighbors, *j,* and connecting *i* and *j* to create a line segment in the feature space. The synthetic samples are produced as a convex combination of the two selected samples *i* and *j*. The following two tables show the percentage of the data for each class of the datasets before and after applying SMOTE.

Table 2. Class distribution of the PAMAP2 dataset

Class	Before Balancing (%)	After Balancing (%)
0 (Rope_jumping)	2.476	8.333
1 (Lying)	10.008	8.333
2 (Sitting)	9.610	8.333
3 (Standing)	9.836	8.333
4 (Walking)	11.955	8.333
5 (Running)	4.978	8.333
6 (Cycling)	8.499	8.333
7 (Nordic_walking)	9.599	8.333
8 (Ascending_stairs)	6.094	8.333
9 (Descending_stairs)	5.458	8.333
10 (Vacuum_cleaning)	9.107	8.333
11 (Ironing)	12.382	8.333

However, as the UCI-HAR dataset owner [35] provides the segmented dataset, applying SMOTE to balance this dataset is inappropriate because SMOTE is applied prior to segmentation. Moreover, we see that the available UCI-HAR segmented dataset is almost balanced as shown in Table 4 below, so it is not so necessary to make it balanced.

Table 3. Class distribution of the WISDM_ar_v1.1 dataset

Class	Before Balancing (%)	After Balancing (%)
0 (Walking)	38.60	16.667
1 (Jogging)	31.20	16.667
2 (Upstairs)	11.20	16.667
3 (Downstairs)	9.10	16.667
4 (Sitting)	5.50	16.667
5 (Standing)	4.40	16.667

Table 4. Class distribution of the UCI-HAR dataset

Class	Class distribution in the available segmented dataset (%)
0 (Walking)	16.625
1 (Walking_upstairs)	15.032
2 (Walking_downstairs)	13.866
3 (Sitting)	17.216
4 (Standing)	18.457
5 (Laying)	18.805

C. Normalizing the Data

The numerical values of different sensors vary considerably (e.g. the numerical values of the magnetometer are much higher than both accelerometer and gyroscope values), training the model with raw sensor data would cause the magnetometer data to have a considerably greater impact than the accelerometer and gyroscope data. Again, as measuring units of different inertial sensors are different (e.g. units of the tri-axial accelerometer (m/s²) and the tri-axial gyroscope (rad/s) differ, resulting in different value ranges. Moreover, raw data may contain noise or outliers due to the sensitivity used while capturing the data. So, data scaling is a critical part of the sensor-based PC-HAR. However, we didn't scale the data in this stage because normalization is handled within the proposed DCNN architecture via a batch normalization layer as the first step.

2.3. Segmenting the Data

To deduce the information the pre-processed data is segmented using a fixed-size sliding window algorithm without inter-window gaps. This algorithm provides a sparse temporal sampling technique and ensures fixed computational cost [16]. Two parameters are determined using domain knowledge to accomplish this: the window size (w) and the shift value (s) [39]. The size of the window generally spans one to 10 seconds of observation data [40]. The shift value describes how much of the data of each window should overlap with the window before. The overlapping method of segmentation provides large quantities of data and deep learning techniques benefit from the availability of large quantities of data. It is also found that due to the higher number of data points for overlapping; the method is less prone to missing important events and the prediction becomes more accurate compared to non-overlapping ones [41]. Increasing the overlapping ratio can improve the recognition accuracy but the memory requirement and the training time will be increased [42]. For labeling in this algorithm in case of multiple labels, often the last label is taken as the label for the complete sequence. Another possibility is to take the majority label.

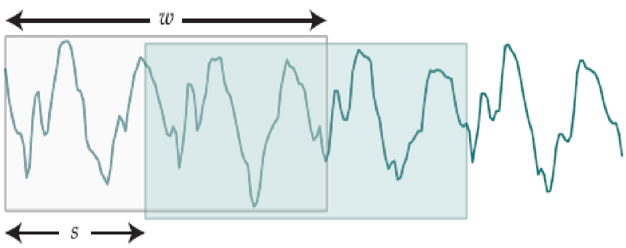


Fig.2. Segmentation process of the proposed method

In this work, we used w = 128 timestamps (which is equivalent to 1.28 seconds of sensor readings for the PAMAP2 dataset, 6.4 seconds of sensor readings for the WISDM_ar_v1.1 dataset, and 2.56 seconds of sensor readings for the UCI-HAR dataset) and overlapping ratio = 50% (i.e. s = 64, the first half of a window contains the observations from the last half of the previous window). The label of each window is selected as the mode of the samples in that window which means the activity data of the majority record within a segment defines the activity class of that segment, which is our target for prediction. Fig. 2 shows the segmentation process of the proposed method using which we obtain a total of 44605 segments for the PAMAP2 dataset, 36951 segments for the WISDM dataset, and 10299 segments (presegmented) for the UCI-HAR dataset.

2.4. Splitting the Segmented Data

We used both holdout and cross-validation methods for splitting the segmented data. For the person-dependent case, we have used the holdout method whereas for the person-independent case, we have used the cross-validation method.

There are several ways in the holdout method to split the data of the datasets, for example: Sample-wise, Subject-wise, and Percentage-wise [43]. We employed a percentage-wise splitting method. In this method, the segmented dataset is split into three subsets in the ratio train: validation: test = 49:21:30. For example, the size of the train, validation, and test subsets of the PAMAP2 segmented dataset are 21856, 9367, and 13382 respectively.

There are several ways in the cross-validation method to split the data of the datasets, for example: K-fold cross-validation, Leave-One-Out Cross-validation, and Leave-One-Group-Out Cross-validation [44]. In this study, the Leave-One-Person-Out Cross-validation (LOPO-CV) scheme is used. In this CV, all data are used for training except data from one subject, which is left for evaluating the classifier. This process is repeated for all available users' data sequences. E.g., in the UCI-HAR dataset, as the number of offered users is 30 it is repeated 30 times for this dataset. Therefore, the test set always contains data from unseen or unknown users. Lastly, the outcomes of these data classifications are aggregated, and the average outcome is considered as the final result.

2.5. DCNN Modeling

The deep model that we have proposed in this work is modified DCNN which is based on [32], developed for the Mcfly automated machine learning package on time series for Python. Varying the hyper-parameters, several different models for DCNN are generated and the best one is selected based on validation accuracy. The predictions obtained from this best model are considered the final predicted activities.

A. Modified DCNN

The high-level architecture of the proposed DCNN is illustrated in Fig. 3. In this architecture, the raw sensor data segment is fed as a 2D matrix to a batch normalization layer which scales the input data, so there is no need to standardize them explicitly in the preprocessing chain stage. The scaled input is passed through one or multiple combinations of the Conv1D layer, batch normalization layer, and ReLU activation function. The benefit of using multiple convolution layers of different filters is that the model can learn local or spatial features of various resolutions. Because of using enough convolutional layers, decreasing the spatial size of the sequence is generally not essential; thus, we do not add pooling layers. Later, the outcomes are flattened and passed through one hidden fully connected layer and ReLU activation function. Finally, single or double fully connected layer(s) and batch normalization layer(s) followed by softmax activation function(s) are used to obtain the final activity class.

The main differences of this modified architecture from the previous model are mainly in the last three (Dense, Batchnorm, and softmax) layers. In the previous model [32], it is proposed to use a single instance of each of the three layers but in this study, we propose to use them either once or twice which depends on the user's participation in activities of the dataset. If each of the subjects performs all the activities then we propose to use the layers for a single time (D = 1) otherwise use them double times (D = 2), this is because a single layer configuration (especially with a fixed dense unit count) is unsuitable for datasets where subjects participate in varying numbers of activities. In the case of D = 2, the first dense layer (and others) uses a fixed value as D = 1, and the second dense layer uses an adjustable value obtained from the number of activities performed by the subject(s). Another mentionable difference is to use of a flattened layer for dimensionality reduction.

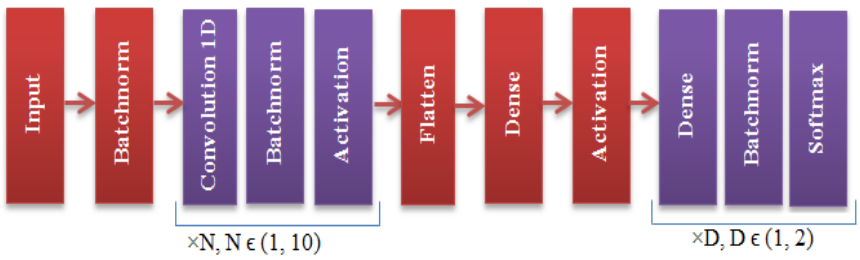


Fig.3. Architecture of the modified DCNN model

Before accomplishing hyper-parameter optimization, we first need to define a parameter grid or search space, where we define a range of potential hyper-parameter values that can be used to develop the model. The number of convolutional layers, the number of filters for each convolutional layer, and the number of neurons in the hidden fully connected layer are the hyper-parameters of this architecture of DCNN which only need to be optimized from a default range during model creation. The number of Convolutional layers of the DCNN is selected in this study from the range [1, 10], the number of filters in each Convolutional layer is selected from the range [10, 100], and the number of neurons in the hidden fully connected layer is selected from the range [10, 2000]. This specific range is defined manually but the effective value from this range is selected using randomized search with cross-validation during model training.

B. Selecting the Best DCNN Model from Multiple Candidates

By altering the hyper-parameters mentioned above, it is possible to obtain many DCNN models. In this study, instead of using exhaustive ‘grid search’, we have used ‘random search’ to generate *G* random models and then train them using the training set, evaluate their performance on the validation set, and select the best model based on validation accuracy. We have used the ‘random search’ optimization algorithm over the ‘grid search’ algorithm because random search is fairly effective and simple [45], and its runtime is significantly lower [46]. In the Grid search algorithm, an exhaustive search is conducted where the model is trained and evaluated on each possible combination of the hyper-parameter values and finally, the model with the best cross-validation result is selected whereas random search only samples and evaluates a random combination of hyper-parameters. The number of generated models, *G*, is experimentally found considering both computational complexity and accuracy. Generating more models helps to find a more accurate model but it costs longer training and validation time whereas the opposite is true in the case of generating few models.

We train all the generated *G* models on the training set for E epochs, with early stopping criteria that the learning procedure will halt when there is no performance progress for successive *S* epochs. For the subject-dependent case, we train the selected model again for more fine-tuning with the training set up to that epoch for which the highest validation result was achieved and without early stopping criteria and lastly assess the model using the test set. For subject-independent case, we train and test that selected model using the LOPO-CV process.

2.6. Evaluating the Model

Accuracy is a decent measure for evaluating our proposed method because data are made balanced in the pre-processing chain stage. Though accuracy is a worthy measure in this study, we have used several other measures and metrics. The performance metrics and measures such as loss curve, accuracy curve, precision, recall, and their harmonic mean, f1-score, as well as learning and prediction time, Cohen’s Kappa score, Jaccard index, phi coefficient, Receiver Operating Characteristics (ROC) curve, etc. are used to evaluate the proposed deep CNN model.

3. Experiments

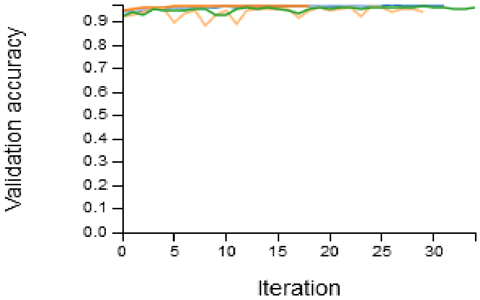
As stated earlier, the experiments of this study are performed on three benchmark datasets: PAMAP2, WISDM_ar_v1.1, and UCI-HAR, each containing on-body sensor data of human activity over time.

3.1. Experimentation Decisions

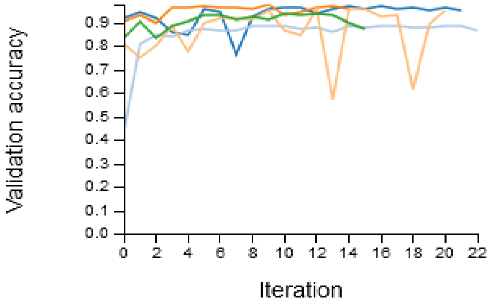
Considering both computational complexity and performance, we see that developing *G* = five models for E = 100 epochs to train and validate those generated models, using S = 5 epochs for early stopping criteria are better selections in this study for implementation.

3.2. Proposed DCNN Model Configuration and Selection

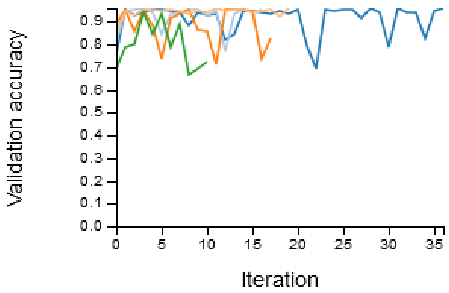
Instead of forming from scratch, we executed the proposed DCNN model and performed its hyper-parameter tuning using the automated machine learning Python library for multivariate time series classification, Mcfly [47]. Each of the five generated DCNN models is trained and validated up to 100 epochs with early stopping criteria. Fig. 4 shows the performance of the five generated models on the validation set for all three HAR datasets. Here we see the variation in epoch numbers among five models because of applying early stopping criteria.



(a) for the PAMAP2 dataset



(b) for the WISDM_ar_v1.1 dataset



(c) for the UCI-HAR dataset

Fig.4. The performances of five generated DCNN models on the validation set for all three HAR datasets

The highest validation accuracies (HVA) of five generated DCNN models and the corresponding epoch number for three HAR datasets are shown in Table 5 below.

Table 5. Highest validation accuracies with the corresponding epoch numbers of five generated models for the datasets

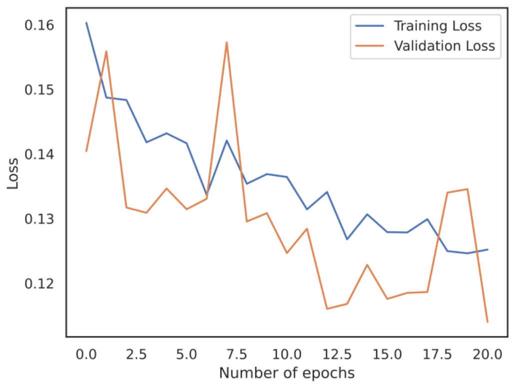
Model	**HVA (epoch) of PAMAP2**	**HVA (epoch) of WISDM_ar_v1.1**	**HVA (epoch) of UCI-HAR**
Model 0	97.22% (27)	97.11% (17)	**95.92% (32)**
Model 1	**97.42% (21)**	88.80% (10)	95.74% (4)
Model 2	97.05% (13)	**97.73% (10)**	95.56% (13)
Model 3	96.46% (25)	95.99% (16)	95.78% (20)
Model 4	96.78% (30)	94.06% (11)	93.97% (4)

From the above table, we see that ‘Model 1’ of PAMAP2, ‘Model 2’ of WISDM_ar_v1.1, and ‘Model 0’ of UCI-HAR provide the highest validation accuracies. So we select those models as the best DCNN model for the corresponding dataset. The number of Conv1D layers, *N*, and that of fully connected layers with batch normalization and softmax activation, *D*, of the aforementioned selected DCNN models with their hyper-parameters are shown in Table 6 for three HAR datasets.

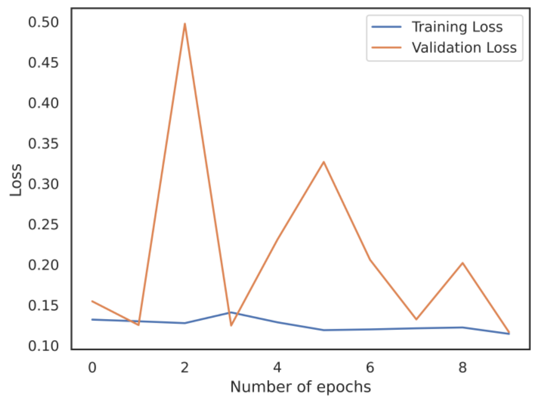
Table 6. Selected DCNN model hyper-parameter values for the datasets

Hyper-parameters	**Best model of PAMAP2 (*N*=6, *D*=1)**	**Best model of WISDM_ar_v1.1 (*N*=9, *D*=2)**	**Best model of UCI-HAR (*N*=10, *D*=1)**
learning_rate	0.002969	0.005252	0.000138
regularization_rate	0.000152	0.000241	0.000862
filters	[66, 14, 36, 84, 20, 69]	[28, 14, 76, 18, 87, 19, 21, 65, 94]	[86, 63, 59, 13, 29, 100, 96, 68, 26, 43]
fc_hidden_nodes	1217	868	305

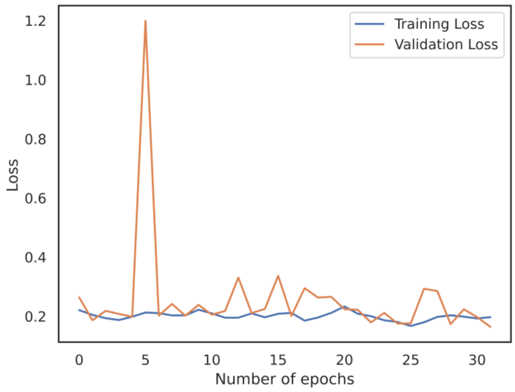
The selected DCNN model is trained again on the whole training set but at this time it is fitted without any early stopping criteria and up to 21, 10, and 32 epochs for the reason that the peak validation accuracies are achieved for these epochs for the corresponding HAR datasets. The training and validation loss curves of the selected DCNN model for three HAR datasets through all epochs are shown in Fig. 5 below.



(a) for the PAMAP2 dataset



(b) for the WISDM_ar_v1.1 dataset



(c) for the UCI-HAR dataset

Fig.5. The training loss and validation loss of the selected DCNN model for three HAR datasets

3.3. *Selected DCNN Model Evaluation*

For the person-dependent case, the selected DCNN model is evaluated on the test dataset. For the person-independent case, the selected DCNN model is evaluated using the LOPO-CV process on the whole dataset. Tables 7, 8, and 9 show the summaries of the selected and trained DCNN models (including model layers and their feature shapes) for the respective datasets.

Table 7. Model summary for the PAMAP2 dataset (layer sequence is from top to bottom and from left to right; Input shape: (None, 128, 39))

Layer Type	**Output Shape**					
BatchNormalization	(None, 128, 39)					
Conv1D	(None, 128, 66)	(None, 128, 14)	(None, 128, 36)	(None, 128, 84)	(None, 128, 20)	(None, 128, 69)
BatchNormalization	(None, 128, 66)	(None, 128, 14)	(None, 128, 36)	(None, 128, 84)	(None, 128, 20)	(None, 128, 69)
Activation	(None, 128, 66)	(None, 128, 14)	(None, 128, 36)	(None, 128, 84)	(None, 128, 20)	(None, 128, 69)
Flatten	(None, 8832)					
Dense	(None, 1217)					
Activation	(None, 1217)					
Dense	(None, 12)					
BatchNormalization	(None, 12)					
Activation	(None, 12)					

Table 8. Model summary for WISDM_ar_v1.1 dataset (layers sequence is from top to bottom and from left to right; Input (None, 128, 3))

Layer Type	**Output Shape**				
BatchNormalization	(None, 128, 3)				
Conv1D	(None, 128, 28)	(None, 128, 14)	(None, 128, 76)	(None, 128, 18)	(None, 128, 87)
BatchNormalization	(None, 128, 28)	(None, 128, 14)	(None, 128, 76)	(None, 128, 18)	(None, 128, 87)
Activation	(None, 128, 28)	(None, 128, 14)	(None, 128, 76)	(None, 128, 18)	(None, 128, 87)
Conv1D	(None, 128, 1)	(None, 128, 21)	(None, 128, 65)	(None, 128, 94)	
BatchNormalization	(None, 128, 19)	(None, 128, 21)	(None, 128, 65)	(None, 128, 94)	
Activation	(None, 128, 19)	(None, 128, 21)	(None, 128, 65)	(None, 128, 94)	
Flatten	(None, 12032)				
Dense	(None, 868)				
Activation	(None, 868)				
Dense	(None, 6)				
BatchNormalization	(None, 6)				
Activation	(None, 6)				

Table 9. Model summary for UCI-HAR dataset (layers sequence is from top to bottom and from left to right; Input (None, 128, 9))

Layer Type	**Output Shape**				
BatchNormalization	(None, 128, 9)				
Conv1D	(None, 128, 86)	(None, 128, 63)	(None, 128, 59)	(None, 128, 13)	(None, 128, 29)
BatchNormalization	(None, 128, 86)	(None, 128, 63)	(None, 128, 59)	(None, 128, 13)	(None, 128, 29)
Activation	(None, 128, 86)	(None, 128, 63)	(None, 128, 59)	(None, 128, 13)	(None, 128, 29)
Conv1D	(None, 128, 100)	(None, 128, 96)	(None, 128, 68)	(None, 128, 26)	(None, 128, 43)
BatchNormalization	(None, 128, 100)	(None, 128, 96)	(None, 128, 68)	(None, 128, 26)	(None, 128, 43)
Activation	(None, 128, 100)	(None, 128, 96)	(None, 128, 68)	(None, 128, 26)	(None, 128, 43)
Flatten	(None, 5504)				
Dense	(None, 305)				
Activation	(None, 305)				
Dense	(None, 6)				
BatchNormalization	(None, 6)				
Activation	(None, 6)				

A. Evaluation of the PAMAP2 Dataset

For the person-dependent case, the proposed DCNN model produced the activity recognition accuracies of 99.18%, 98.21%, and 98.11% for the PAMAP2 training, validation, and test sets, respectively. The accuracy is calculated by dividing the number of correctly recognized samples by the total number of samples in the respective set. Fig. 6 shows the ROC curve of the DCNN model for the PAMAP2 test dataset. The Area Under the Curve (AUC) is 1.0 for five out of twelve classes. For the remaining classes, the AUC is approximately 1.0 (0.99). Moreover, the macro-average ROC curve area is 0.99, indicating the robustness of the model on the entire PAMAP2 test set.

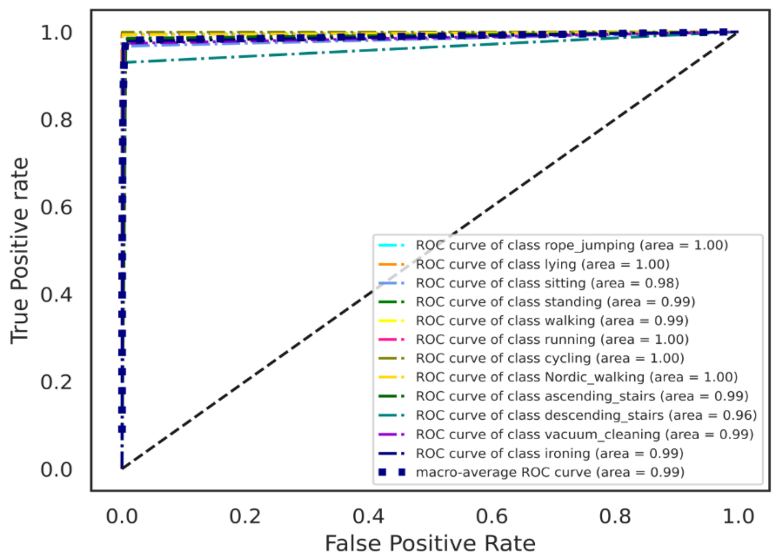


Fig.6. ROC curve (one vs. rest) of the proposed DCNN model for the PAMAP2 dataset

For the person-independent case, the proposed DCNN model achieved an average training accuracy of 96.15% and an average validation accuracy of 90.27% using the LOPO-CV process for the PAMAP2 dataset. Fig. 7 shows the training and validation accuracy curves of this process through all subjects for this dataset.

B. Evaluation of the WISDM_ar_v1.1 Dataset

For the percentage-wise random data splitting case, the selected DCNN model produced the activity recognition accuracies of 99.76%, 98.53%, and 98.48% for the WISDM_ar_v1.1 training, validation, and test sets, respectively. Fig. 8 below shows the ROC curve of our model for this dataset and except for the 'Upstairs' and 'Downstairs' activity classes, the AUC for the remaining four activity classes is 1.0. Since 'Upstairs' and 'Downstairs' are inverse activities, some misclassifications occurred between them. Moreover, the macro-average ROC curve area is approximately 1.0 (0.99), indicating the robustness of the model on the entire WISDM_ar_v1.1 test set.

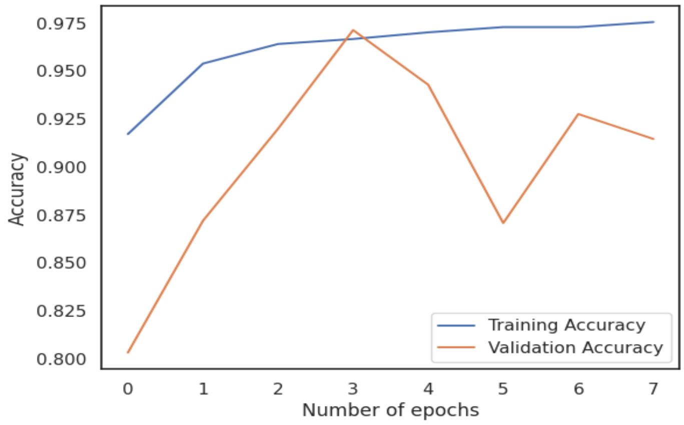


Fig.7. The training accuracy and validation accuracy of the LOPO-CV process for the PAMAP2 dataset

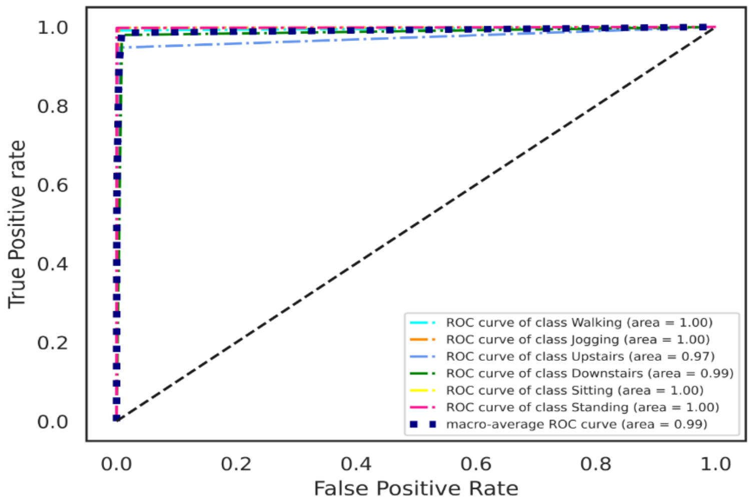


Fig.8. ROC curve (one vs. rest) of the proposed DCNN model for the WISDM dataset

For the person-independent case, the selected DCNN model achieved an average training accuracy of 96.53% and an average validation accuracy of 94.51% using the LOPO-CV process for the WISDM dataset. Fig. 9 shows the training and validation accuracy curves of this process through all subjects for this dataset.

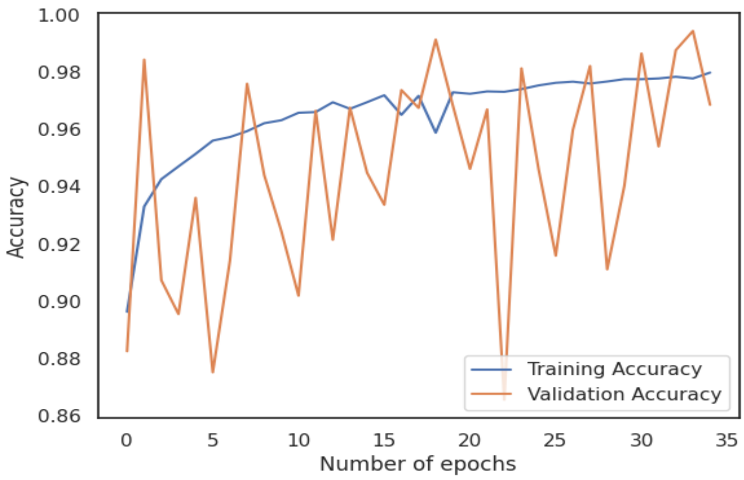


Fig.9. The training accuracy and validation accuracy of the LOPO-CV process for the WISDM dataset

C. Evaluation of the UCI-HAR Dataset

For the person-dependent case, the proposed DCNN model achieved activity recognition accuracies of 96.25%, 95.56%, and 93.25% for the UCI-HAR training, validation, and test sets, respectively. Fig. 10 below shows the ROC curve of the proposed model for this dataset and two out of six classes are perfectly classified (AUC of 1.0 and 0.99), while the AUC for the remaining four classes is lower. The macro-average ROC curve area is 0.96 for the entire UCI-HAR test dataset.

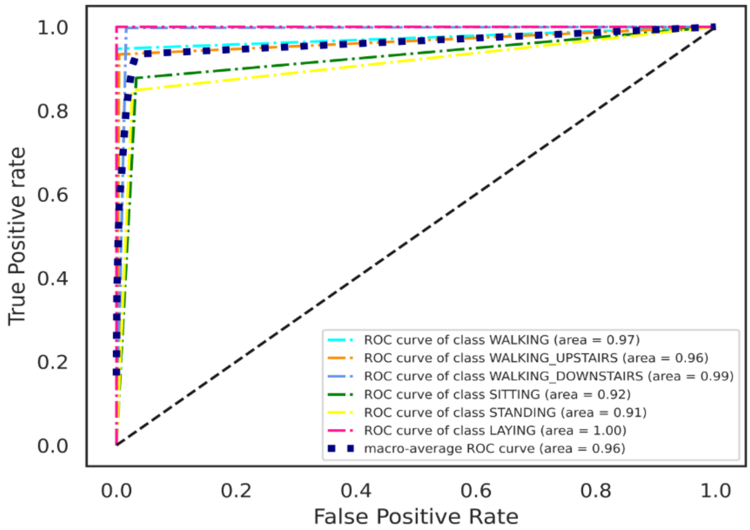


Fig.10. ROC curve (one vs. rest) of the proposed DCNN model for the UCI-HAR dataset

For the person-independent case, the proposed model achieved an average training accuracy of 98.89% and an average validation accuracy of 98.67% using LOPO-CV for the UCI-HAR dataset. Fig. 11 shows the training and validation accuracy curves of this process through all subjects for this dataset.

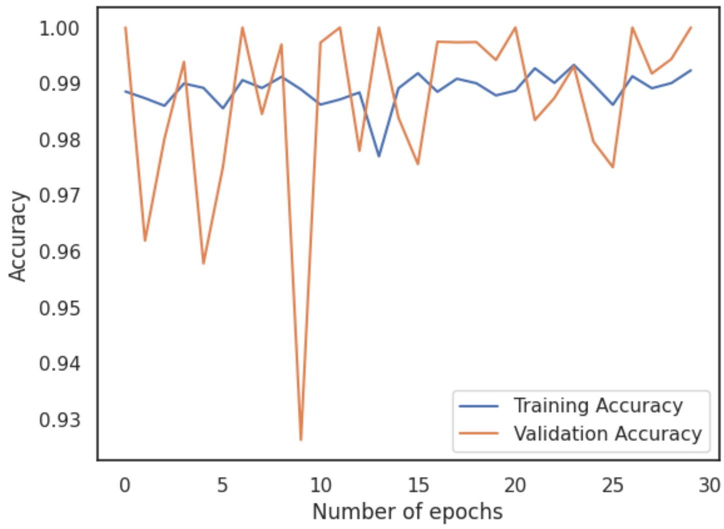


Fig.11. The training accuracy and validation accuracy of the LOPO-CV process for the UCI-HAR dataset

4. Analysis and Comparison

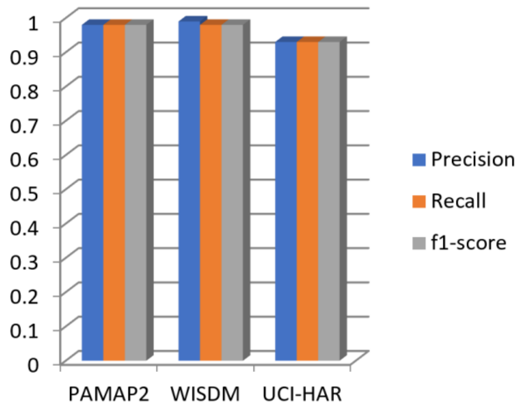
In this section, we assess the proposed method using various evaluation measures beyond accuracy, analyze the execution time, and perform comparisons with state-of-the-art methods.

4.1. Assessment using Other Performance Measures

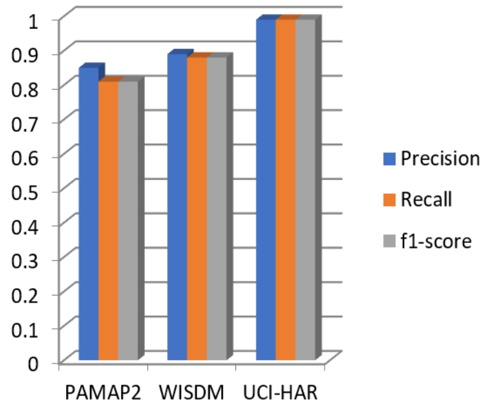
To assess our model, though accuracy is a worthy performance measure because of data being balanced in our pre-processed datasets, we also report results using other metrics, including precision, recall, and F1-score as well as Cohen's kappa, Jaccard index, and phi coefficient.

A. Assessment using Precision, Recall, and f1-score Measures

The precision for any activity class α is the number of correctly predicted instances of α divided by the total number of predicted instances of α. The recall for any activity class α is the number of correctly predicted instances of α divided by the total number of actual instances of α. The f1-score of any activity, *α*, is the harmonic mean of the precision and recall of this activity [44]. These values for both person-dependent and person-independent cases for three HAR datasets are shown in Fig. 12 below.



(a) for person-dependent case



(b) for person-independent case

Fig.12. The precision, recall, and f1-score values for three HAR datasets

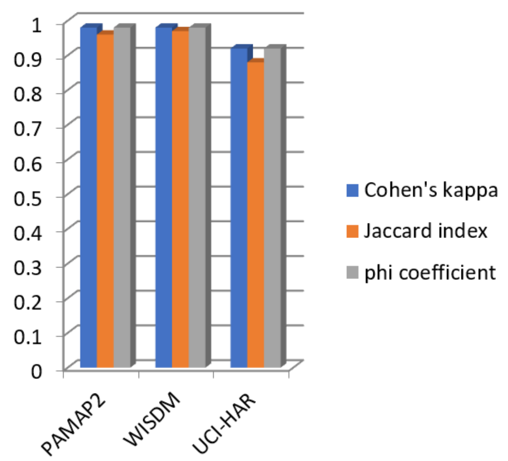
As shown in Fig. 12(a), for the person-dependent case, the proposed model achieved macro-average precision values of 0.98, 0.99, and 0.93; macro-average recall values of 0.98, 0.98, and 0.93; and macro-average F1-scores of 0.98, 0.98, and 0.93 for the three HAR datasets, respectively. These values are obtained by macro-averaging the corresponding metrics for individual activity classes. As shown in Fig. 12(b), for the LOPO-CV case, the proposed model achieved macro-average precision values of 0.85, 0.89, and 0.99; macro-average recall values of 0.81, 0.88, and 0.99; and macro-average F1-scores of 0.81, 0.88, and 0.99 for the three datasets, respectively.

The minimal variation between precision and recall values for each class and across the three HAR datasets in both cases indicates the balanced performance of the proposed model.

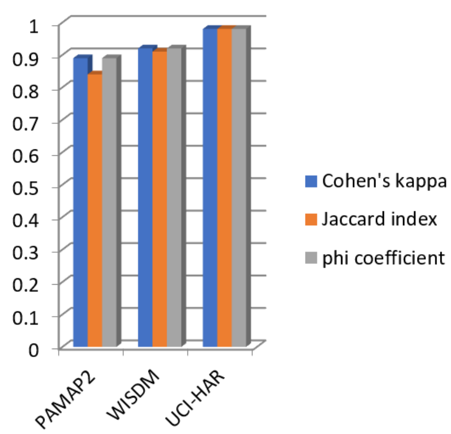
B. Assessment using Cohen's Kappa, Jaccard Index, and Phi Coefficient Measures

Cohen's kappa measures the agreement between observed and expected classifications. Its value lies between -1 and 1. A value above 0.8 is usually considered a good score. The Jaccard index measures the similarity between the predicted and actual label sets. The phi coefficient, equivalent to the Matthews correlation coefficient (MCC) for binary classification, is a value between -1 and +1. A value of +1 indicates a perfect recognition [44]. These values for both person-dependent and person-independent cases for three HAR datasets are shown in Fig. 13 below.

As shown in Fig. 13(a), for the person-dependent case, the proposed model achieved Cohen's kappa values of 0.98, 0.98, and 0.92; Jaccard index values of 0.96, 0.97, and 0.88; and phi coefficient values of 0.98, 0.98, and 0.92 for the HAR datasets, respectively. As shown in Fig. 13(b), for the LOPO-CV case, the proposed model achieved Cohen's kappa values of 0.89, 0.92, and 0.98; Jaccard index values of 0.84, 0.91, and 0.98; and phi coefficient values of 0.89, 0.92, and 0.98 for the three datasets, respectively.



(a) for person-dependent case



(b) for person-independent case

Fig.13. Cohen's kappa, Jaccard index, and phi coefficient for three HAR datasets

The results from these performance measures corroborate the conclusions drawn from the accuracy analysis.

4.2. Time Complexity

The learning and prediction time of the proposed model for three HAR datasets for both person-dependent and person-independent cases are presented in Table 10. We observed that the learning and prediction times are reasonable for all datasets, particularly considering the complexity of the deep model, making it suitable for real-time HAR applications. Moreover, inference time for a single data sample is significantly faster.

Table 10. Learning and prediction times (HH:MM:SS:ms) for the datasets

		PAMAP2	WISDM_ar_v1.1	UCI-HAR
All five models' learning time		0:46:41.738799	0:24:12.550994	0:09:59.961810
Person-dependent case	Selected DCNN model's learning time	0:03:25.901769	0:01:57.356518	0:01:53.465518
	Selected DCNN model's prediction time	0:00:02.879076	0:00:02.630435	0:00:01.358650
Person-independent case	LOPO-CV time	0:03:55.908207	0:13:15.547512	0:04:36.980201

Early stopping was employed during training to halt the process if no improvement in performance was observed for five consecutive epochs. Consequently, training rarely reaches the maximum of 100 epochs, resulting in reduced training times for both the candidate and selected models despite their depth. The training time for the five candidate models ranged from approximately 10 to 47 minutes across the three datasets, whereas the selected model required at most 3.5 minutes. The prediction time is at most less than only 3 milliseconds for the whole test dataset. This demonstrates the favorable trade-off between computational efficiency and performance in the proposed method.

4.3. Comparison with Related Methods

A comparison of the proposed method with related HAR methods on these standard datasets is outlined in Table 11.

Table 11. Comparison between the proposed DCNN method and the related methods for the datasets

Dataset	Year [Ref.]	Model/Classifier	Accuracy	Person dependency
PAMAP2	2019 [10]	Inception GoogLeNet-GRU	93.5%	
	2020 [48]	CNN	91%	
	2020 [48]	SVM	84.07%	
	2021 [49]	CNN-BiLSTM	94.27%	
	2021 [50]	Multi-input CNN-GRU	95.27%	
	2021 [51]	MLP-D	87.9%	
	2022 [5]	Ensem-HAR	97.45%	
	2022 [21]	MLWP	95.93%	
	2022 [52]	Condconv CNN	94.01%	
	2023 [53]	DL + SoTA	94.91%	
	2023 [54]	GRU-INC	90.30%	
	2023 [55]	DLT	98.52%	
	2024	**DCNN (Proposed)**	**98.11%**	**Person-dependent**
	2024	**DCNN (Proposed)**	**90.27%**	**Person-independent**
WISDM_ar_v1.1	2019 [56]	Adaboost using Naïve Bayes	86.72%	
	2019 [57]	Very Fast Decision Tree	85.90%	
	2020 [58]	LSTM-CNN	95.85%	
	2020 [11]	EnsemConvNet	97.2%	
	2020 [59]	Linear SVM	89.90%	
	2021 [50]	Multi-input CNN-GRU	97.21%	
	2022 [60]	AMC-CNN	95.18%	
	2022 [5]	Ensem-HAR	98.70%	
	2023 [61]	ConvBiLSTM-GRU	97.12%	
	2023 [62]	GRU + attention	97.90%	
	2023 [55]	DLT	97.90%	
	2024 [63]	Voting of Ensembles	86.67%	
	2024	**DCNN (Proposed)**	**98.48%**	**Person-dependent**
	2024	**DCNN (Proposed)**	**94.51%**	**Person-independent**

UCI-HAR	2020 [9]	ANN	96.30%	
	2021 [50]	Multi-input CNN-GRU	96.20%	
	2021 [22]	Linear SVC	96.56%	
	2022 [5]	Ensem-HAR	95.05%	
	2022 [21]	MLWP	97.71%	
	2023 [61]	ConvBiLSTM-GRU	97.23%	
	2023 [62]	GRU + attention	96.00%	
	2023 [54]	GRU-INC	96.27%	
	2023 [53]	DL + SoTA	97.11%	
	2023 [64]	Stacking	97.12%	
	2024 [65]	FusionActNet	97.35%	
	2024 [37]	Ensem-DeepHAR	99.51%	
	2024	**DCNN (Proposed)**	**93.25%**	**Person-dependent**
	2024	**DCNN (Proposed)**	**98.67%**	**Person-independent**

As shown in the comparison table, the proposed DCNN method achieves the highest accuracy for the person-dependent case and lower accuracy for the person-independent case compared to most previous works for PAMAP2 and WISDM datasets. Conversely, for the UCI-HAR dataset, the proposed method shows lower accuracy for the person-dependent case but the highest accuracy for the person-independent case compared to most previous works.

5. Conclusions

This paper presents a new deep learning-based HAR method employing the modified DCNN model and body-worn sensor data. Extensive data analysis and preprocessing were carried out using PC-HAR. Data segmentation was performed on the pre-processed data using a fixed-size, overlapping sliding window technique. The model is designed to extract multi-resolution features using filters of varying sizes, requiring fine-tuning of only a few hyperparameters. Wearable sensor data from three publicly accessible benchmark datasets were used for the experiments in this work. The experimental results revealed that the proposed DCNN model performed effectively on these HAR datasets. Though a high similarity between some activities leads to wrong prediction in some cases, the model demonstrates commendable performance, achieving higher accuracy than numerous recent approaches applied to the revealed HAR datasets. This DCNN-based HAR method, however, has drawbacks. It did not simultaneously outperform state-of-the-art methods in both person-dependent and person-independent cases. Moreover, there is room for improvement in recognizing complex activities and those with similar patterns. These aspects represent directions for future work.

All the Declarations and Statements

Author Contributions Statement

S.M. Mohidul Islam – Conceptualization, Methodology, Proposed research ideas, constructed the overall framework, and implementing the research model: dataset preprocessing, Model Training, Validation, and Performance Evaluation, validated results using standard metrics, and benchmarked performance against existing methods, Writing – Drafted the initial manuscript.
Kamrul Hasan Talukder – Data Curation and Supervision, Formal Analysis, Visualization, and Statistical Analysis: Performed in-depth analysis of experimental results, and supervised project execution, Writing – Review and Editing, helped coordinate project milestones and deadlines.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This research is funded by the Information and Communication Technology division of the Posts, Telecommunications, and Information Technology Ministry, the government of Bangladesh through the ICT fellowship.

Data Availability Statement

This study analyzed publicly available datasets. The datasets can be found here: PAMAP2 dataset: "https://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring", WISDM dataset:

“https://www.cis.fordham.edu/wisdm/dataset.php”, and UCI-HAR dataset: “https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphone”, accessed on “05/01/2025”.

Ethical Declarations
The study didn’t involve any human subjects directly and/or animals.

Acknowledgments
None

Declaration of Generative AI in Scholarly Writing
None

Abbreviations
The following abbreviations are used in this manuscript:

DCNN – Deep Convolutional Neural Network
LOPO-CV – Leave One Person Out Cross-Validation
PC-HAR – Preprocessing Chain for Human Activity Recognition

APPENDIX
None

References

[1] K. Sidney. Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living. Journal of the American Geriatrics Society, 31(12):721-727, December 1983. https://doi.org/10.1111/j.1532-5415.1983.tb03391.x.
[2] K. Ranjit and V. Geetha. Human activity recognition in smart home using deep learning techniques. In 2021 13th International Conference on Information & Communication Technology and System (ICTS), pages 230-234, October 2021. https://doi.org/10.1109/ICTS52701.2021.9609044.
[3] A.K.M. Masum, S. Jannat, E.H. Bahadur, M.G.R. Alam, S.I. Khan, and M.R. Alam. “Human Activity Recognition Using Smartphone Sensors: A Dense Neural Network Approach,” in 2019 International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), May 2019. https://doi.org/10.1109/ICASERT.2019.8934657.
[4] X. Zhou, W. Liang, K. Wang, H. Wang, L. T. Yang, and Q. Jin, Deep-learning-enhanced human activity recognition for Internet of healthcare things. IEEE Internet of Things Journal, 7(7):6429-6438, April 2020.
[5] B. Debarshi, D. Sharma, W. Kim, M.F. Ijaz, and P.K. Singh. Ensem-HAR: An ensemble deep learning model for smartphone sensor-based human activity recognition for measurement of elderly health monitoring. Biosensors, 12(6):393, June 2022. https://doi.org/10.3390/bios12060393.
[6] G. Walid, R. Elbasiony, and S. Ashry. ADL classification based on autocorrelation function of inertial signals. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 833-837, December 2017. https://doi.org/10.1109/ICMLA.2017.00-53.
[7] M. Stikic, T. Huynh, K.V. Laerhoven, and B. Schiele. ADL recognition based on the combination of RFID and accelerometer sensing. In 2008 second international conference on pervasive computing technologies for healthcare, pages 258-263, January 2008. https://doi.org/10.1109/PCTHEALTH.2008.4571084.
[8] A. Baldominos, P. Isasi, and Y. Saez. Feature selection for physical activity recognition using genetic algorithms. In 2017 IEEE Congress on Evolutionary Computation (CEC), pages 2185-2192, June 2017. https://doi.org/10.1109/CEC.2017.7969569.
[9] B.A. Atalaa, A. Alenany, A. Helmi, and I. Ziedan. Effect of data segmentation on the quality of human activity recognition. East Afr. Scholars J. Eng. Comput. Sci., 3(7):133–145, 2020. https://doi.org/10.36349/easjecs.2020.v03i07.001.
[10] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan. InnoHAR: A deep neural network for complex human activity recognition. IEEE Access, 7:9893-9902, January 2019. https://doi.org/10.1109/ACCESS.2018.2890675.
[11] D. Mukherjee, R. Mondal, P.K. Singh, R. Sarkar, and D. Bhattacharjee. EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. Multimedia Tools and Applications, 79:31663-31690, November 2020. https://doi.org/10.1007/s11042-020-09537-7.
[12] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang. Deep residual bidir-LSTM for human activity recognition using wearable sensors. Mathematical Problems in Engineering, vol. 2018, p. 7316954, 2018. https://doi.org/10.1155/2018/7316954.
[13] E. Gambi, G. Temperini, R. Galassi, L. Senigagliesi, and A. De Santis, ADL recognition through machine learning algorithms on IoT air quality sensor dataset. IEEE Sensors Journal, 20(22):13562-13570, June 2020. https://doi.org/10.1109/JSEN.2020.3005642.
[14] W. Hom. Human action recognition with still images and video [online]. Available at: https://github.com/wendell-hom/human_action_recognition (Accessed on 5 January 2024).
[15] S.A. Bux, X. Wang, P. Angelov, and Z. Habib. Human action recognition using transfer learning with deep representations. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 463-469, May 2017. https://doi.org/10.1109/IJCNN.2017.7965890.
[16] Y. Abdulazeem, H. M. Balaha, W. M. Bahgat, and M. Badawy, “Human action recognition based on transfer learning approach,” IEEE Access, vol. 9, pp. 82058–82069, 2021, doi: 10.1109/ACCESS.2021.3083888.

[17] N. Hao, D. Nguyen, K. Yonekawa, M. Kurokawa, S. Wada, and K. Yoshihara. Multi-source transfer learning for human activity recognition in smart homes. In 2020 IEEE international conference on smart computing (SMARTCOMP), pages 274-277, September 2020. https://doi.org/10.1109/SMARTCOMP50058.2020.00063.
[18] Z. Shibo, Y. Li, S. Zhang, F. Shahabi, S. Xia, Y. Deng, and N. Alshurafa. Deep learning in human activity recognition with wearable sensors: A review on advances. Sensors, 22(4):1476, February 2022. https://doi.org/10.3390/s22041476.
[19] J. Yang, Y. Xu, H. Cao, H. Zou, and L. Xie, Deep learning and transfer learning for device-free human activity recognition: A survey. Journal of Automation and Intelligence, 1(1):100007, December 2022.
[20] B. Sefen, B. Sebastian, D. Andreas, and A. Slim, Human activity recognition-using sensor data of smartphones and smartwatches. International Conference on Agents and Artificial Intelligence, 2:488-493, SciTePress, February 2016.
[21] B. Zhou, C. Wang, Z. Huan, Z. Li, Y. Chen, G. Gao, H. Li, C. Dong, and J. Liang, A novel segmentation scheme with multi-probability threshold for human activity recognition using wearable sensors. Sensors, 22(19):7446, September 2022.
[22] W. Kong, L. He, and H. Wang, Exploratory data analysis of human activity recognition based on smartphone. IEEE Access, 9:73355-64, May 2021.
[23] Khan, A.M., Siddiqi, M.H. and Lee, S.W. Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. Sensors, 13(10):13099-13122, September 2013.
[24] N. Y. Hammerla, S. Halloran, and T. Plötz, “Deep, convolutional and recurrent models for human activity recognition using wearables,” arXiv preprint arXiv:1604.08880, April 2016.
[25] W. Shi, X. Fang, G. Yang, and J. Huang, “Human activity recognition based on multichannel convolutional neural network with data augmentation,” IEEE Access, 10:76596-76606, July 2022.
[26] Ihianle, I.K., Nwajana, A.O., Ebenuwa, S.H., Otuka, R.I., Owa, K. and Orisatoki, M.O. A deep learning approach for human activities recognition from multimodal sensing devices. IEEE Access, 8:179028-179038, September 2020.
[27] S. Gupta. Deep learning based human activity recognition (HAR) using wearable sensor data. International Journal of Information Management Data Insights, 1(2):100046, November 2021. https://doi.org/10.1016/j.jjimei.2021.100046.
[28] R.K. Athota, and D. Sumathi. Human activity recognition based on hybrid learning algorithm for wearable sensor data. Measurement: Sensors, 24:100512, December 2022. https://doi.org/10.1016/j.measen.2022.100512.
[29] A. Gumaei, M. M. Hassan, A. Alelaiwi, and H. Alsalman, A hybrid deep learning model for human activity recognition using multimodal body sensing data. IEEE Access, 7:99152-60, August 2019.
[30] P. Compagnon, L. Grégoire, D. Stefan, and G. Christophe, Learning personalized ADL recognition models from few raw data. Artificial Intelligence in Medicine, 107:101916, July 2020.
[31] X. Shi, Y. Li, F. Zhou, and L. Liu, Human activity recognition based on deep learning method. In 2018 International Conference on Radar (RADAR), pages 1-5, August 2018.
[32] D.V. Kuppevelt, C. Meijer, F. Huber, A.V.D. Ploeg, S. Georgievska, and V.T.V. Hees. Mcfly: Automated deep learning on time series. SoftwareX, 12:100548, July 2020. https://doi.org/10.1016/j.softx.2020.100548.
[33] A. Reiss, and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In 2012 16th International Symposium on Wearable Computers (ISWC), pages 108-109, June 2012. https://doi.org/10.1109/ISWC.2012.13.
[34] J.R. Kwapisz, G.M. Weiss, and S.A. Moore. Activity Recognition using Cell Phone Accelerometers. ACM SigKDD Explorations Newsletter, 12(2):74-82, March 2011. https://doi.org/10.1145/1964897.1964918.
[35] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. “A public domain dataset for human activity recognition using smartphones,” in Proceedings of the 21st European Symposium on Artificial Neural Networks (ESANN), pp. 3–4, Apr. 2013.
[36] Glen, S. Data Analysis & Exploratory Data Analysis (EDA) [online]. Available at: https://www.statisticshowto.com/probability-and-statistics/data-analysis/ (Accessed on 1 September 2022).
[37] Islam, S.M. and Talukder, K.H. Ensem-DeepHAR: Identification of Human Activity in Smart Environments using Ensemble of Deep Learning Methods and Motion Sensor Data. Measurement: Sensors, 36:101398, December 2024.
[38] H. He and Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications. Hoboken, NJ, USA: Wiley, 2013.
[39] J. O. Laguna, A. García, and D. Borrajo, A Dynamic SlidingWindow Approach for Activity Recognition. In Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization (UMAP), Girona, Spain, 11–15 July 2011, pages 219–230, 2011.
[40] Preece, S.J., Goulermas, J.Y., Kenney, L.P. and Howard, D. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. IEEE Transactions on Biomedical Engineering, 56(3):871-879, October 2008.
[41] C.-T. Yen, J.-X. Liao, and Y.-K. Huang, “Human daily activity recognition performed using wearable inertial sensors combined with deep learning algorithms,” IEEE Access, vol. 8, pp. 174105–174114, 2020, doi: 10.1109/ACCESS.2020.3025938.
[42] Baldominos, A., Cervantes, A., Saez, Y. and Isasi, P. A comparison of machine learning and deep learning techniques for activity recognition using mobile devices. Sensors, 19(3):521, January 2019.
[43] M. Bock, A. Hölzemann, M. Moeller, and K. Van Laerhoven, Improving deep learning for HAR with shallow LSTMs. In Proceedings of the 2021 ACM International Symposium on Wearable Computers, pages 7-12, September 2021.
[44] Scikit-learn [online]. Available at: https://scikit-learn.org/stable/modules/cross_validation.html (Accessed on 1 February 2024).
[45] Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. The Journal of machine learning research, 13(1):281-305, February 2012.
[46] Comparing randomized search and grid search for hyperparameter estimation [online], Available at: https://scikit-learn.org/stable/auto_examples/model_selection/plot_randomized_search.html (Accessed on 10 January 2024).
[47] Mcfly [online], Available at: https://mcfly.readthedocs.io/en/latest/ (Accessed on 10 January 2024).
[48] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu. Deep learning models for real-time human activity recognition with smartphones. Mobile Networks and Applications, 25(2):743-755, April 2020. https://doi.org/10.1007/s11036-019-01445-x.
[49] S.K. Challa, A. Kumar, and V.B. Semwal. A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data. The Visual Computer, 38(12):4095-4109, December 2022. https://doi.org/10.1007/s00371-021-02283-3.
[50] N. Dua, S.N. Singh, and V.B. Semwal. Multi-input CNN-GRU based human activity recognition using wearable sensors. Computing, 103(7):1461-1478, July 2021. https://doi.org/10.1007/s00607-021-00928-8.
[51] Jin, L., Wang, X., Chu, J. and He, M. Human activity recognition machine with an anchor-based loss function. IEEE Sensors

Journal, 22(1):741-756, November 2021.
[52] Cheng, X., Zhang, L., Tang, Y., Liu, Y., Wu, H. and He, J. Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices. IEEE Sensors Journal, 22(6):5889-5901, February 2022.
[53] Challa, S.K., Kumar, A., Semwal, V.B. and Dua, N. An optimized deep learning model for human activity recognition using inertial measurement units. Expert Systems, 40(10):e13457, December 2023.
[54] Mim, T.R., Amatullah, M., Afreen, S., Yousuf, M.A., Uddin, S., Alyami, S.A., Hasan, K.F. and Moni, M.A. GRU-INC: An inception-attention based approach using GRU for human activity recognition. Expert Systems with Applications, 216:119419, April 2023.
[55] Ige, A.O. and Noor, M.H.M. A deep local-temporal architecture with attention for lightweight human activity recognition. Applied Soft Computing, 149:110954, December 2023.
[56] Subasi, A., Fllatah, A., Alzobidi, K., Brahimi, T. and Sarirete, A. Smartphone-based human activity recognition using bagging and boosting. Procedia Computer Science, 163:54-61, January 2019.
[57] Zhang, Y. and Ramachandran, K.M. Human activity recognition with streaming smartphone data. In 2019 Global Conference for Advancement in Technology (GCAT), pages 1-6, October 2019.
[58] K. Xia, J. Huang, and H. Wang. LSTM-CNN architecture for human activity recognition. IEEE Access, 8:56855-56866, March 2020. https://doi.org/10.1109/ACCESS.2020.2982225.
[59] Zhang, Y. and Ramachandran, K.M. Offline machine learning for human activity recognition with smartphone. 2020.
[60] M. Jaberi and R. Ravanmehr, “Human activity recognition via wearable devices using enhanced ternary weight convolutional neural network,” Pervasive and Mobile Computing, vol. 83, p. 101620, 2022, doi: 10.1016/j.pmcj.2022.101620.
[61] Sain, M.K., Singha, J., Saini, S. and Semwal, V.B. Human action recognition using ConvBiLSTM-GRU in indoor environment. In 2023 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), pages 179-186, December 2023.
[62] Wang, Y., Xu, H., Liu, Y., Wang, M., Wang, Y., Yang, Y., Zhou, S., Zeng, J., Xu, J., Li, S. and Li, J. A novel deep multifeature extraction framework based on attention mechanism using wearable sensor data for human activity recognition. IEEE Sensors Journal, 23(7):7188-7198, February 2023.
[63] Islam, S.M. and Talukder, K.H. Ensem-EnsemHAR: Human Activity Recognition using Ensemble of Ensemble Methods and Smartphone Sensor Data. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pages 1-7, June 2024.
[64] Islam, S.M. and Talukder, K.H. Exploratory analysis of smartphone sensor data for human activity recognition. IEEE Access, 11:99481-99498, September 2023.
[65] U. Saha, S. Saha, M. T. Kabir, S. A. Fattah, and M. Saquib, “Decoding human activities: Analyzing wearable accelerometer and gyroscope data for activity recognition,” IEEE Sensors Letters, vol. 8, no. 7, 2024.

Authors’ Profiles

S. M. Mohidul Islam obtained his Ph.D. degree in Computer Science and Engineering (CSE) in 2025 from Khulna University, Bangladesh. He achieved his B.Sc. Engineering degree in CSE with distinction in 2007. He completed his M.Sc. Engineering with distinction in 2016 and achieved an ICT fellowship from Bangladesh Government for his M.Sc. Research. He joined Khulna University as a faculty member in 2008. He has published several research papers in international journals and conferences. His research interests include human activity recognition, data science, machine learning, and smart technology. Mr. Islam is a life member of the Engineer’s Institution Bangladesh (IEB) and Bangladesh Computer Society (BCS). He is the former Joint secretary of the Khulna Region of Bangladesh Open Source Network (BdOSN). He is the former Joint secretary and current Branch member of the Khulna Branch of Bangladesh Computer Society.

Dr. Kamrul Hasan Talukder obtained his Bachelor of Science degree in CSE with distinction. He completed his M.Sc. in Computer Science from the National University of Singapore (NUS) in 2004 and his Doctor of Engineering (D. Eng.) from Hiroshima University, Japan in 2008. He is a professor of Computer Science and Engineering Discipline at Khulna University. He is the former Dean of the Science, Engineering, and Technology School of Khulna University. He served the Computer Science and Engineering Discipline as the head of the Discipline for 3 years. He joined Khulna University as a faculty member in 2000. Dr. Talukder has published more than 70 peer-reviewed research articles over the years. His research interests are image analysis, software engineering, networking, IoT, etc. He was a Japan Society for the Promotion of Science (JSPS) postdoctoral fellow for 2 years at Hiroshima University.

How to cite this paper

S. M. Mohidul Islam, Kamrul Hasan Talukder, "Modeling Human Activity Recognition with Deep Convolutional Neural Networks and Wearable Sensor Data", International Journal of Intelligent Systems and Applications(IJISA), Vol.18, No.4, pp.97-115, 2026. DOI:10.5815/ijisa.2026.04.06