


RESEARCH ARTICLE

Hybrid deep learning model-based human action recognition in indoor environment

Manoj Kumar Sain¹ , Rabul Hussain Laskar², Joyeeta Singha¹ and Sandeep Saini¹

¹Department of Electronics and Communication, The LNM Institute of Information Technology, Jaipur, India and

²Department of Electronics and Communication, National Institute of Technology, Silchar, Assam, India

Corresponding author: Manoj Kumar Sain; Email: manoj.sain@lnmiit.ac.in

Received: 18 March 2023; **Revised:** 10 August 2023; **Accepted:** 31 August 2023; **First published online:** 10 October 2023

Keywords: Human Activity Recognition (HAR); Convolutional Neural Network; Deep Learning; LSTM; BiLSTM; Kinect V2 sensor

Abstract

Human activity recognition (HAR) is an emerging challenge among researchers. HAR has many possible uses in various fields, including healthcare, sports, and security. Furthermore, there are only a few publicly accessible datasets for classifying and recognizing physical activity in the literature, and these datasets comprise fewer activities. We created and compared our dataset with available datasets, that is, NTU-RGBD, UP-FALL, UR-Fall, WISDM, and UCI HAR. The proposed dataset consists of seven activities: eating, exercise, handshake, situps, vomiting, headache, and walking. The activities were collected from 20 people between the ages of 25 and 40 years using Kinect V2 sensor at 30 FPS. For classification, we use deep learning architectures based on convolutional neural network (CNN) and long short-term memory (LSTM). Additionally, we developed a novel hybrid deep learning model by combining a CNN, a bidirectional LSTM unit, and a fully connected layer for activity identification. The suggested model builds unique guided features using the preprocessed skeleton coordinates and their distinctive geometrical and kinematic aspects. Results from the experiment are contrasted with the performance of stand-alone CNNs, LSTMs, and ConvLSTM. The proposed model's accuracy of 99.5% surpasses that of CNN, LSTM, and ConvLSTM, which have accuracy rates of 95.76%, 97%, and 98.89%, respectively. Our proposed technique is invariant of stance, speed, individual, clothes, etc. The proposed dataset sample is accessible to the general public.

1. Introduction

Human activity recognition (HAR) is a process of automatically recognizing the activity of an individual or group. HAR is an emerging challenge among researchers. HAR has many possible uses in a variety of fields, including healthcare [1], sports [1], and security [1]. The area is widely used in various applications such as video surveillance in the military, human–computer interaction, sports, etc. [2]. There are two methods for identifying human activity: one relies on wearable sensors and visual sensors. A sensor that can record the body's acceleration, angular velocity, gravity, etc., is attached to the body in the wearable-based system. In the vision-based system, an RGB camera captures the subject's activities. Various joint information is gathered from each frame with depth-based sensors [3]. An ensemble model is proposed for local correlation that makes use of intra-class variability and class center reliability. The suggested pedestrian attention module aids in focusing on certain features, while the proposed priority-distance graph convolutional network (PDGCN) module predicts class center nodes and determines distances [4]. AFCDL, an adaptive fusion, and category-level dictionary learning paradigm, has been proposed to solve drawbacks such as shifting angles of view, background clutter, and movement patterns. AFCDL enhances sample reconstruction for action recognition by learning adaptive weights for each camera. With a 3% to 10% gain in recognition accuracy across four multiview action benchmarks, it trumps cutting-edge techniques.

Microsoft Kinect and other low-cost, high-mobility sensors are widely used for recognizing human motion [1]. Several techniques have been developed that use the information from the skeleton joint to identify human indications. Computer vision researchers have paid particular attention to Kinect's ability to monitor skeleton joints. Elements are independent of a person's size, look, and changing camera angles that can be extracted using the skeleton joint feature of the Kinect [2]. Recognition of activity with conventional cameras may be complex due to illumination variations, stance variations, and cluttered backgrounds. Kinect can record motion information even in changing illumination, poses, and complex environments.

If we talk about the skeleton joint's features, it cannot identify human activities. Some modifications or adding more features are required for efficient activity classification. The gain in momentum in center of mass coordinates, velocity, and acceleration, as well as other derived features like a person's height and the different joints angles, and distance between joints, can be calculated for identification during actions like walking, vomiting, situps, or differentiate between similar motion activities, etc. Such features are programmatically engineered because they cannot be directly calculated by convolution neural network and long short-term memory (LSTM) architecture [5] approaches. In this study, the Kinect-generated features are inputted into a deep learning network and a collection of hand-engineered kinematic features. Using a Kinect V2 sensor, we first applied skeletal tracking algorithms and gathered 3D joint locations for each frame. To improve the model's performance, a new set of features, that is, distance feature, different joints angle, etc., have been created with the help of Kinect-generated features. Natural body coordinates have been utilized to calculate velocity, acceleration, and joint angle properties. The dataset is preprocessed once the features are extracted and fed into the deep learning models. The proposed model is a fusion model of CNN and bidirectional long short-term memory (BiLSTM). In the proposed model, the output of each convolutional block is added to the following convolutional block output to prevent the network from vanishing gradient and to improve the feature quality. In the proposed model, input feature vector passes through two feature extraction branches: on one side, features are inputted to the CNN and BiLSTM layer series and generate new feature vectors. Conversely, the input feature vector passes through the BiLSTM layer to generate a new feature vector. Finally, these spatial and temporal features are concatenated and inputted into a fully connected network for a probabilistic classification score. The manuscript's significant contributions are as follows:

1. A dataset named "LNMIIT-KHAD (LNMIIT-Kinect Human Activity Dataset)" has been developed from 20 individuals and has seven indoor activities such as eating, walking, headache, handshake, situps, exercise, and vomiting with all possible variations.
2. A hybrid deep learning classification model comprising CNNs and BiLSTM Layers has been proposed. The model can extract spatiotemporal features from the input feature sets to classify human activities efficiently and precisely. The model also consists of a dropout and regularizing layer to improve the model performance and prevent the model from overfitting.
3. A comparative analysis has been done for different deep learning models with the LNMIIT-KHAD dataset. The model has been evaluated on publicly available datasets to check the model performance.
4. To protect user privacy, a set of auto-generated features and hand-engineered kinematic features has been used as an input feature vector for the proposed hybrid deep learning model instead of the visual input.

The paper consists of five sections. Section 2 is related to the literature survey. Section 3 is associated with the dataset description. Section 4 describes the proposed methodology, which includes data collection and preprocessing, data normalization and scaling, feature selection, windowing and segmentation, and the proposed classification model. Section 5 describes experimental results followed by references.

2. Related Works

Existing literature [6, 7] used inertial measurement unit (IMU)-enabled devices on their waists, wrists, and feet to capture human activity data in the form of acceleration, angular velocity, and other metrics. Before further processing, several preprocessing steps, such as noise reduction and normalization, have been applied to raw data. The human behavior is then classified using a feature extraction and model training procedure. Several methods for simulating and recognizing human actions have been presented. Early researchers mainly employed decision trees, support vector machines (SVMs), naive Bayes, and other machine learning methods to identify the data gathered by IMU sensors. Researchers have developed classification models for recognizing human activity using various methods. For instance, Wang et al. [8] presented a DF network, a profoundly fully connected network to attain an exemplary data structure collated with a synthetic neural network. The original skeleton graphs were converted by Ke et al. [9] into pseudo-frames in four human body areas, and CNN was used to extract spatial attributes from the pseudo-structures. Additionally, Liu et al. [10] proposed that by encoding skeletal joints into spatial and temporal volumes, a three-dimensional convolutional neural network (CNN) was used to collect spatial and time sequence information at a local timescale. By considering the three-dimensional geometric relationship between the human parts that used body part rotations and its translations in 3-dimensional space, Vemulapalli et al. [11] proposed that new skeleton representation has been presented. The authors classified human activities using Lie algebra and described human actions as curves in the Lie group. Scana et al. [12] development of an automated system for motor assessment of individuals with neurological disorders used the Kinect sensor. They assessed the reaching performance scale score derived from the Kinect data, and the results were comparable to the visual score derived in a clinical setting. Vishwanath et al. [13] have proposed a methodology for recognizing human activity using human gait patterns. They used an IMU sensor with three degrees of freedom to capture seven different activities. They also introduced kinematic features and Kinect-generated features for classifying activities. Rahul Jain et al. [14] proposed a methodology to achieve the walking pattern classification. For that, human lower extremity activities are considered to understand walking behavior. An IMU has been used as a wearable device to capture the walking movement of different lower limb joints. For activity classification, two different deep learning models, namely CNN and LSTM, have been used. Vishwanath et al. [15] proposed a hybrid deep learning approach for post-stroke rehabilitation. Microsoft Kinect V2 has been used to capture the targeted activities. Different combinations of deep learning models have been used for classification; CNN-Gated recurrent unit (GRU) achieved the highest accuracy. Vijay Bhaskar Semwal et al. [16] proposed multitasking human walking activity recognition using human gait patterns. IMU sensor has been used to capture different walking patterns of the candidates. Various combinations of deep learning models have been used for activity classification; GRU-CNN achieved the highest accuracy. Nidhi Dua et al. [17] proposed a multi-input hybrid deep learning model for HAR. Wearable sensors like gyroscopes and accelerometers have been used to collect human activity data. The model achieved a classification accuracy of 95%. Santosh et al. [18] proposed a deep architecture fusion of CNN and LSTM. The final model achieved an accuracy of 98% for a self-collected dataset.

The author uses principal component analysis to reduce feature dimensions by extracting Euclidean distance and spherical coordinates between normalized joints. HAR is carried out using statistical characteristics and principal component analysis, proposed in ref. [19], and globally contextualized attention LSTM [20]. To tackle the sensor-based HAR challenges, ref. [21] presented a multilayer ResGCNN (graph convolutional neural network) residual structure. The deep transfer learning tests utilizing the ResGCNN model demonstrate excellent transferability and few-shot learning performance. Table I lists the dataset utilized in this study. Some of the works related to HAR have been listed in Table I.

3. Proposed methodology

To track and identify human activity in the interior environment, we conducted extensive ablation research, created a new dataset, and created a novel hybrid model based on deep learning. In the first

Table I. Related work-based human action recognition system.

Related work	Sensor used	Dataset used	Classifier	Accuracy
M. Zeng et al. [6]	Wearable	Opportunity	CNN	88.19%
X. Jiang et al. [22]	Mobile (Smart) Phone	UCI	Convolutional Neural Network	97.5%
M. Gholamrezai et al. [23]	Mobile (Smart) Phone	UCI	CNN	95.69%
S. Dhanraj et al. [7]	Mobile (Smart) Phone	UCI	CNN	93.93%
A. Adedin et al. [24]	Mobile (Smart) Phone Wearable	a. Wireless Sensor-Data Mining (WISDM); b. OPPORTUNITY	CONVAE	94%; 84.9%
S. Yu and L. Qin [25]	Mobile (Smart) Phone	UCI	BiLSTM	93.79
Y. Zaho et al. [26]	Mobile (Smart) Phone Wearable	1. UCI; 2. OPPORTUNITY	Residual BiLSTM	93.6%
S. Deep et al. [27]	Smartphone	UCI	CNN-LSTM	93.40%
K. Xia et al. [28]	Mobile (Smart) Phone Body-Worn	a. UCI; b. WISDM; c. OPPORTUNITY	LSTM-CNN	95.78%; 95.85%; 92.63%
Y. Yan et al. [21]	9-D right waist; 9-D left ankle; 9-D back	a. PAMAP2; b. mHealth; c. TNDA	HAR-ResGCNN	97.86%; 96.95%; 99.10%

step, the activity is recorded with the help of the Kinect V2 sensor at 30 FPS. There is a possibility of null value detection due to no activity recorded for that time. As per our algorithm for a null value, it shows no activity. The recorded data may also contain some outliers. The linear interpolation method has been used to remove outliers. The feature set has been normalized to reduce duplication using the 3D joints method indicated in Eq. (1). Table II shows details about Kinect V2 sensor specifications. In addition to Kinect-generated information, certain valuable features have been retrieved for recognizing various activities, such as velocity, acceleration, the angle between joints, a person's height, and the distance between joints. Next, a set of selected features is made out of Kinect-generated and kinematic features. Finally, the modified input feature sets are shaped according to the deep learning classification models

Table II. Kinect motion sensor V2 specification.

S. No.	Parameter	Value
1	Device name	Microsoft Kinect V2
2	Sampling rate	30Hz
3	Type	RGBD
4	Frame size	1920 × 1080 pixels
5	Connectivity	3.0 USB
6	Field of view	70 deg. × 60 deg.
7	Skeleton joints defined	25
8	Minimum skeletal tracking	6
9	Operating measuring range	0.5 m to 4.5 m

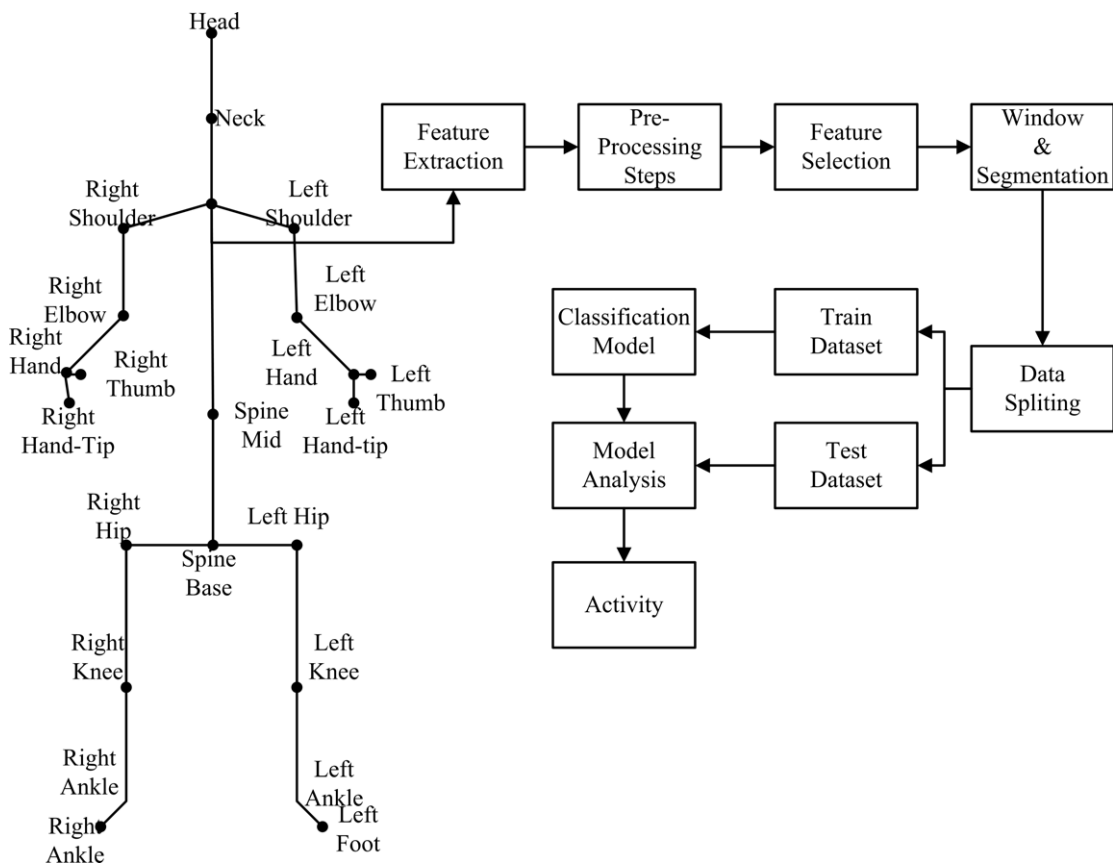


Figure 1. Proposed methodology.

and input to the state-of-the-art deep learning models for classification score generation. Figure 1 shows the proposed methodology:

$$X_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{1}$$

- X_{norm} = normalized value of a feature value
- x_{min} = minimum value of a feature vector
- x_{max} = maximum value of a feature vector

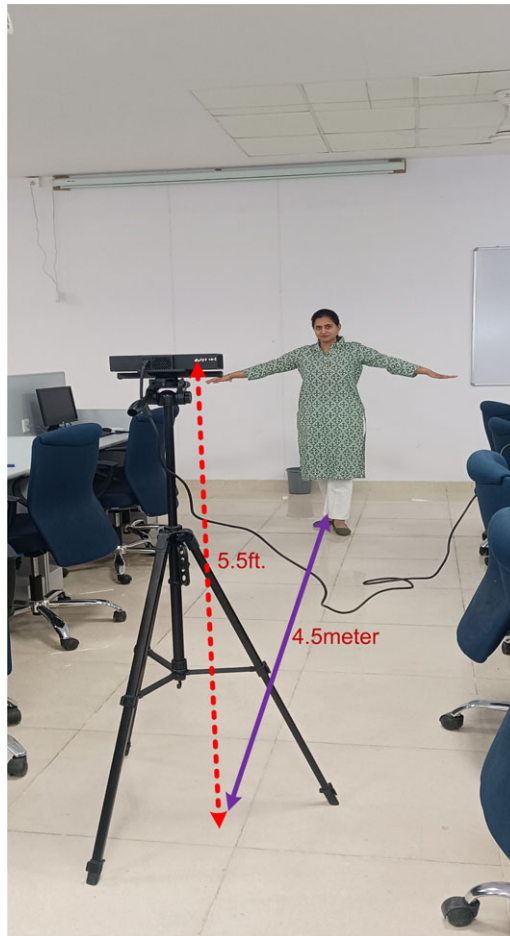


Figure 2. Dataset recording setup.

3.1. Dataset collection and preprocessing

An in-depth examination of the data gathering and improvement process is provided in this section. We created our own dataset and presented it in this publication. A set for dataset collection has been shown in Fig. 2. The proposed dataset consists of seven activities, that is, walking, situp, eating, handshaking, headache, exercise, and vomiting, collected from 20 people (13 males and 7 females) between the ages of 25 and 40 years. Microsoft Kinect sensor V2 can identify 25 distinct joints in the human body. We extract camera and orientation coordinate values from every joint in the human body and save them in comma separated values (CSV) files. We gathered data from 20 distinct volunteers, each completing a task for 20 s, as shown in Table III. Each task has 400 samples, with each participant doing it 10 times (70 samples per participant). The files for each activity are concatenated.

After assembling all the necessary data, the raw data are subjected to different preprocessing techniques. Only a few of the skeleton's joints are instructive for a given task, as ref. [29] noted, so not all skeletal joints are useful. In our situation, we omitted joints such as the Spine Base, Head, Shoulder Left, Elbow Left, Hand Left, Shoulder Right, Elbow Right, Hand Right, Hip Left, Knee Left, Ankle Left, Foot Left, Hip Right, Knee Right, Ankle Right, and Foot Right as shown in Fig. 3 that are not crucial for identifying the intended activity. Table IV describes the list of tracked skeleton joints, a set of Kinect-generated features, derived features, and activity class labels. Some evaluated images of State of the art (SOTA) datasets, that is, WISDM, UCI-HAR, and NTU-RGBD datasets, are shown in Fig. 4. For

Table III. Dataset description.

Labels	Activity name	Participants	Time/activity (s)	FPS	Sample/activity
1	Eating	20	20	30	400
2	Exercise	20	20	30	400
3	Handshake	20	20	30	400
4	Headache	20	20	30	400
5	Situps	20	20	30	400
6	Vomiting	20	20	30	400
7	Walking	20	20	30	400

model evaluation on SOTA datasets, MediaPipe framework has been used to extract the joint's motion information as shown in Fig. 5.

3.1.1 Windowing and segmentation

Windowing and segmentation are used in many HAR applications [30]. Segmentation is typically used during the preprocessing stage to facilitate data analysis. Windowing is a frequent segmentation technique. The sampling frequency used in Kinect V2 activity recording is 30 Hz (30 samples per second). Figure 6 depicts data splitting into an allotted frame (Window) of 2.57 s (80 attributes set) with a 0.5 intersection.

3.2. Geometric and Kinematic feature calculation

The coordinates of the joints in the 3D human skeleton are utilized to evaluate various features and build feature vectors. Feature vectors are programmatically calculated by using Kinect V2 features generated for each frame. These are the unique features that easily distinguish each activity with one another. As per the dataset, only 16 joints are selected for activity classification as mentioned in Fig. 3.

3.2.1 Angle between skeleton joints

An illustrated skeleton is created by connecting the 3D coordinates of the various body joints with a line. As per our dataset, six relevant joints, namely elbow right, elbow left, knee left, knee right, hip left, and hip right, are utilized for the angle feature calculation. Figure 7 presents a distance and angle calculation method. The average difference between the hip-to-knee and ankle-to-knee values is used for calculating the angle value. If joint1 is the hip joint, joint2 is the knee joint, and joint3 is the ankle joint, and then the angle between skeletons is as follows:

$$\Theta = \frac{\text{joint1joint2joint3}}{\text{joint1joint2} * \text{joint2joint3}} \quad (2)$$

$$\text{joint1joint2joint3} = \text{joint1}_1 * \text{joint1}_2 + \text{joint2}_1 * \text{joint2}_2 + \text{joint3}_1 * \text{joint3}_2 \quad (3)$$

$$\text{joint1}_1 = x1 - y1, \text{joint2}_1 = x2 - y2, \text{joint3}_1 = x3 - y3 \quad (4)$$

$$\text{joint1joint2} = \sqrt{\text{joint1}_1^2 + \text{joint2}_1^2 + \text{joint3}_1^2} \quad (5)$$

$$\text{joint2joint3} = \sqrt{\text{joint1}_2^2 + \text{joint2}_2^2 + \text{joint3}_2^2} \quad (6)$$

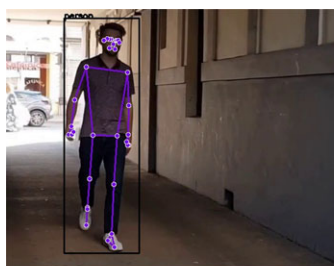
$$\text{Angle} = \frac{\cos^{-1} \theta * 180}{\pi} \quad (7)$$

Activity	Frame Sequence Represent Movement of Relative Joint		Joints Name
Eating			<ol style="list-style-type: none"> 1. Spine Base 2. Head 3. Shoulder Left 4. Elbow Left 5. Hand Left 6. Shoulder Right 7. Elbow Right 8. Hand Right 9. Hip Left 10. Knee Left 11. Ankle Left 12. Foot Left 13. Hip Right 14. Knee Right 15. Ankle Right 16. Foot Right
Exercise			
Handshake			
Headache			
Situps			
Vomiting			
Walking			

Figure 3. Joints details as per activity.

Table IV. Set of features specifications.

S. No.	Label	Elucidation
1	Microsoft Kinect V2 skeleton joints attributes	Elbow (L-R), head, hands (L-R), shoulder (L-R), spine base, hip (L-R), knee (L-R), ankle (L-R), and foot (L-R)
2	Features	Orientation Y, orientation X, orientation Z, camera X, camera Y, and camera Z
3	Derived features	Velocity in X, Y, and Z directions, angle at elbows, angle at knees, angle at hips, height of the person, distance between joints, average velocity, acceleration in X, Y, and Z directions
4	Class labels	Eating, exercise, handshake, headache, situps, vomiting, and walking



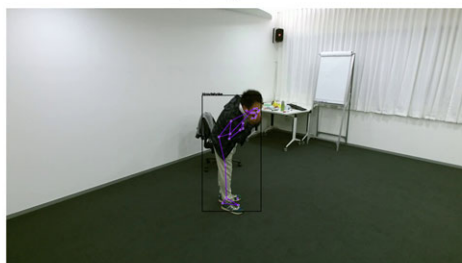
Walking



Sitting



Handshake



Vomiting

Figure 4. Samples from WISDM, UCI-HAR, and NTU-RGBD datasets.

3.2.2 Velocity prediction

The velocities of the X, Y, and Z axes are calculated using the difference between the human skeleton's coordinates at time t and time $t + 1$. Along with measuring velocity, the differences between succeeding frames are also utilized to detect acceleration in the X, Y, and Z axes. We included average acceleration and average velocity as a characteristic as well. In a HAR application that detects human gestures or body movements, velocity can be utilized to distinguish between slow and fast motions and recognize certain movement patterns. Analyzing velocity patterns allows users to distinguish between different activities or gestures, detect irregularities or sudden changes in motion, and characterize movement speed. It can be calculated as:

$$\text{Velocity}(t) = \text{distance between joints} / \text{time Elapsed} \quad (8)$$

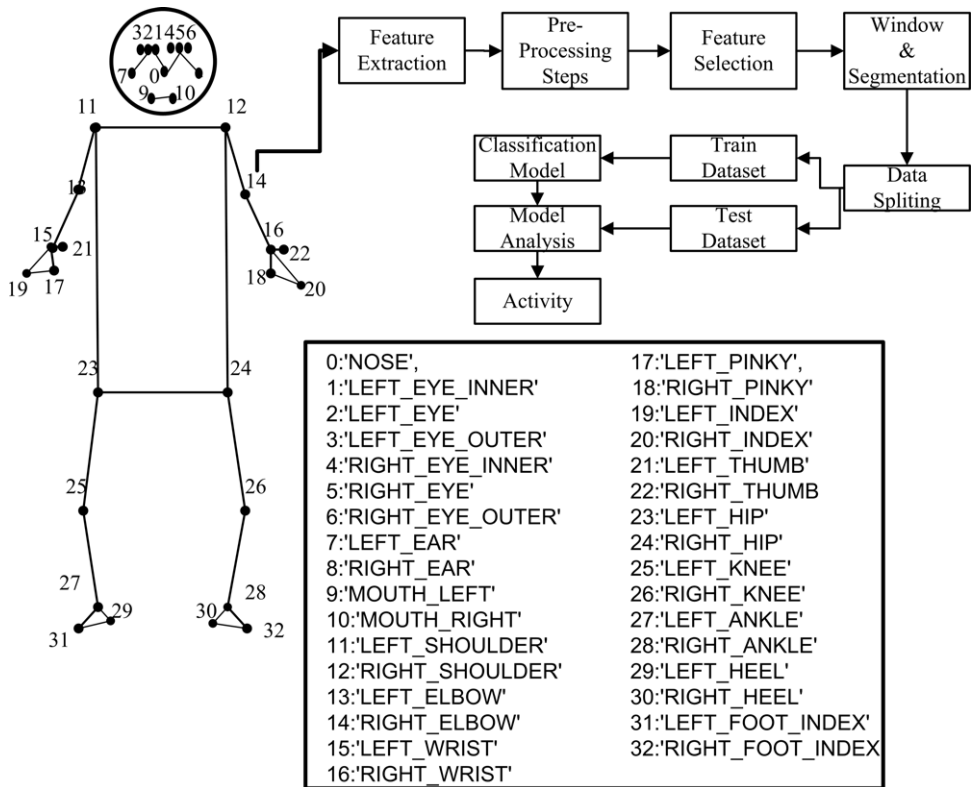


Figure 5. Classification process for NTU-RGBD, UP-FALL, and UR-Fall datasets.

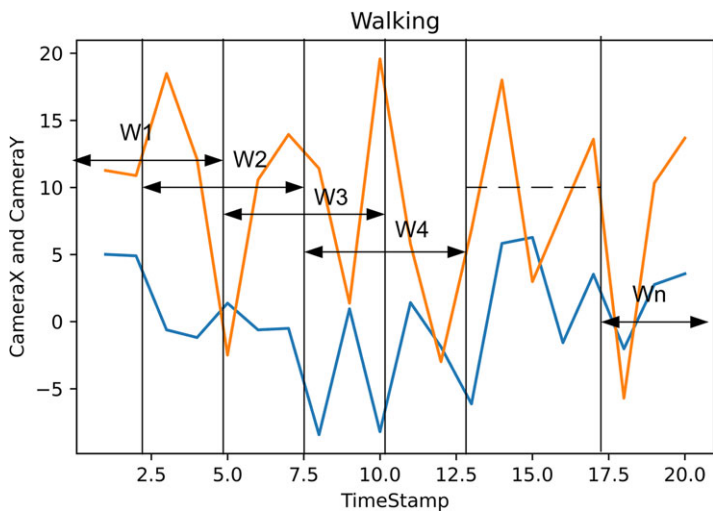


Figure 6. Windowing and segmentaion.

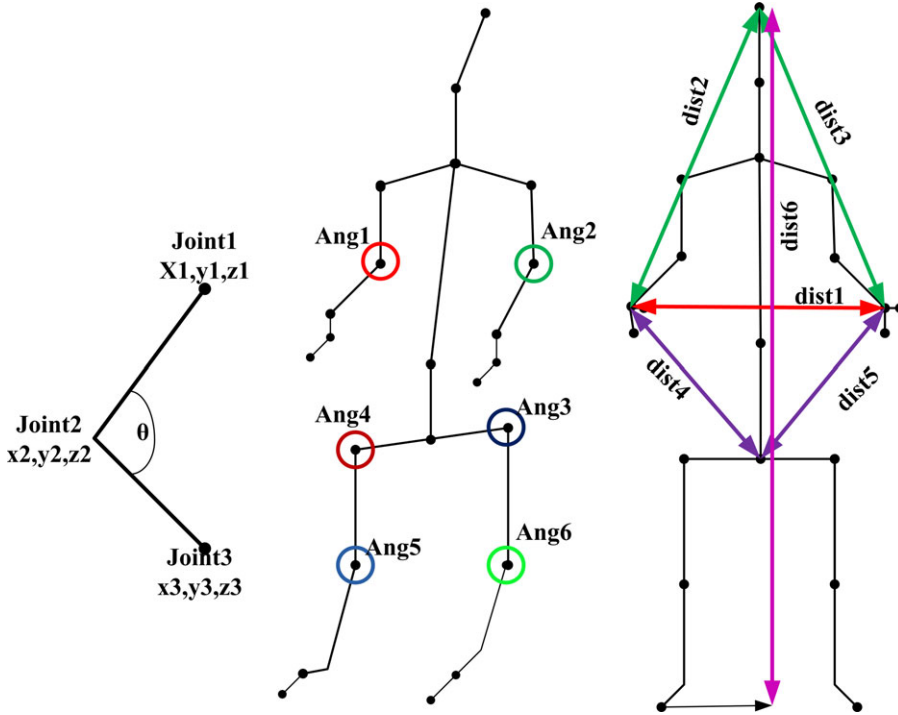


Figure 7. Geometric and kinematic features.

$$v(t) = \frac{\sqrt{(\text{Joint}_x(t+1) - \text{Joint}_x(t))^2 + (\text{Joint}_y(t+1) - \text{Joint}_y(t))^2}}{(t+1) - t} \tag{9}$$

where $\text{Joint}_x(t+1)$ is value of position x of respective joint at $t+1$;
 $\text{Joint}_y(t+1)$ is value of position y of respective joint at $t+1$;
 $\text{joint}_x(t)$ is value of position x of respective joint at t ;
 $\text{Joint}_y(t)$ is value of position t of respective joint at t .

3.2.3 Distance prediction

Through the dataset, we also included a displacement feature vector that included the distances between the hands, between the hand and the head, and the distance between the hand and the spine base:

$$\text{Distance}(t) = \sqrt{(\text{Joint}_x(t+1) - \text{Joint}_x(t))^2 + (\text{Joint}_y(t+1) - \text{Joint}_y(t))^2} \tag{10}$$

3.2.4 Height prediction

The distance between the extreme joints at the top and bottom of the body determines the height. Because every person’s height varies, this factor normalizes other feature vectors.

3.3. Classification models

The completed dataset and its derived features are fed into the deep learning network in this stage for classification. Different classification techniques have been employed, including CNN, LSTM, ConvLSTMs, and the proposed hybrid deep neural network.

1. **Convolutional Neural Network Architecture** [22]: First, the activity recognition has been implemented using the CNN [22]. Let X_t be the 1D feature vector consisting of Kinect-generated and derived features. The convolutional layers output can be given as:

$$\text{Out}_i^{l,j} = \sigma \left(\text{Bi}_j + \sum_{m=1}^M \text{wt}_m^j * X_{i+m-1}^{0j} \right) \tag{11}$$

where l is the layer index and σ is the sigmoid activation function. Bi_j is the bias associated with the j th feature, and M is the filter size. wt is the weight for the $j^{\text{A normalization}}$ and the m th to normalize the input values and lead to more accurate activation. Kernel regularizer and dropout layers are used to minimize the overfitting of the model.

2. **Long Short-Term Memory Architecture** [31]: Second, we have implemented activity recognition using LSTM, an improved version of recurrent neural network (RNN), which avoids the vanishing gradient problem and consists of memory cells. A single-cell, three-gate LSTM module can selectively learn, unlearn, or retain knowledge from each entity. LSTM's cell state facilitates an uninterrupted flow of information between units by allowing a few linear exchanges. Each component has inputs, outputs, and forget gates that can add or remove data from the cell state. The forget gate uses a sigmoid function to choose whether to ignore information from previous cell states. The input gate regulates the flow of information about the current cell state by performing point-wise multiplication of "sigmoid" and "tanh." The output gate determines which data have to be sent at the conclusion.

Studies have been done on the impact of different batch sizes, hidden layers, and learning rates. Two stacked LSTM layers with 100 neurons each for 7 classes, a learning rate of 0.0025, and a batch size of 128 yielded the best results. The Adam optimizer was used to calculate losses using the softmax loss function.

3. **ConvLSTM Architecture** [32]: Next, we applied ConvLSTM [32] to classify our dataset. CNN and LSTM were combined to create ConvLSTM. Here, CNN was used to extract spatial characteristics, LSTM to predict sequences, and dense layers to map the features to create a more separable space. The hyperparameters had been optimized for the size, number of layers, steps, batch size, and learning rate (0.0001). The shape of the feature vector is first set to a 3D tensor, including batch shape, steps, and input feature dimension. Then after the convolution operation, features again reshape to the 3D array as (batch size, time steps, and sequence length). We used Adam for optimization. The model got the precision and F1 score of 97.89% and 97.75%.

4. **Proposed Hybrid Deep Learning Architecture**: We proposed the hybrid approach in which parallel feature learning methodology has been used. The input features vector has been applied to the 1D convolution and BiLSTM layers. The architecture shown in Fig. 8 has two parallel paths. One path has layers of 1D convolution layer, 1D max-pooling layers, and a BiLSTM layer. The path extracts both spatial and temporal features. In this path, input features are also added to the spatial features extracted from convolution layers. The path extracts spatial and temporal features without losing any input characteristic. The second path extracts only temporal features. These features are concatenated into a single feature vector. The combined feature has been passed through a BiLSTM layer. They were finally flattening the features. Two dense classification layers and the softmax activation function generate a probabilistic classification of activities. Table V shows the proposed hybrid deep learning model summary:

$$y_t = w_y * h_t \tag{12}$$

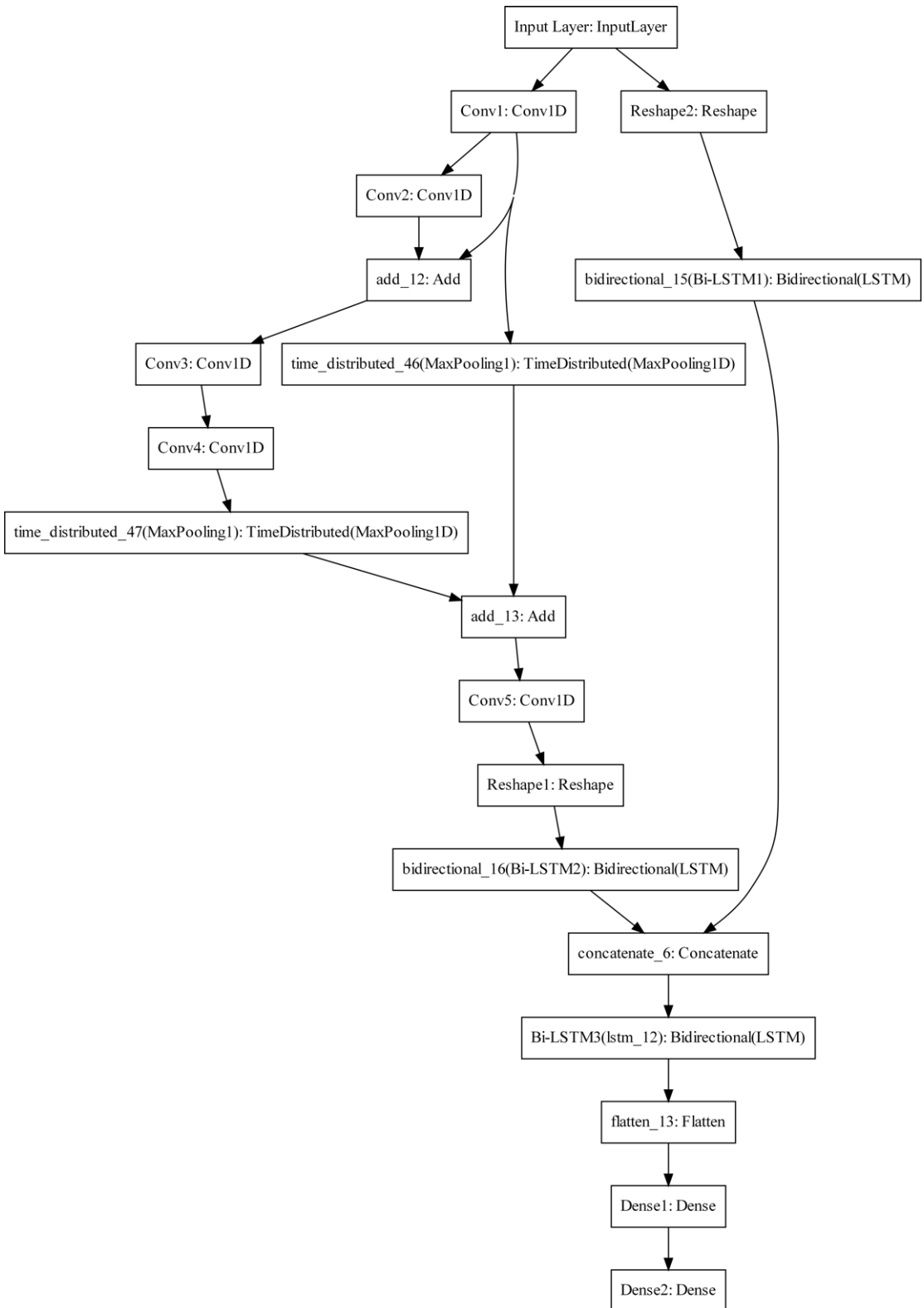


Figure 8. Proposed classification model.

Table V. Proposed hybrid deep learning model summary.

Layer number	Layer name	Layer type	Layer shape
0	Input Layer	Input Layer	(None, None, 40, 13)
1	1D Convolution Layer	Convolution	(None, None, 39, 32)
2	1D_Convolution_Layer_2	Conv1D	(None, None, 39, 32)
3	Add_layer1	Add	(None, None, 39, 32)
4	1D_Convolution_Layer_3	Conv1D	(None, None, 38, 32)
5	1D_Convolution_Layer_4	Conv1D	(None, None, 38, 32)
6	time_distributed_24	TimeDistributed	(None, None, 19, 32)
7	time_distributed_25	TimeDistributed	(None, None, 19, 32)
8	Add_layer2	Add	(None, None, 19, 32)
9	1D_Convolution_Layer_5	Conv1D	(None, None, 18, 32)
10	Reshape_Layer_2	Reshape	(None, None, 520)
11	Reshape_Layer_1	Reshape	(None, None, 576)
12	BiLSTM_Layer_1	Bidirectional	(None, None, 200)
13	BiLSTM_Layer_2	Bidirectional	(None, None, 200)
14	Concatenation_Layer	Concatenate	(None, None, 400)
15	BiLSTM_Layer_3	Bidirectional	(None, 200)
16	Flatten_Layer	Flatten	(None, 200)
17	Dense_Layer_1	Dense	(None, 64)
18	Dense_Layer_2	Dense	(None, 7)

4. Experimental results

The experimental findings employing CNNs, LSTMs, ConvLSTM, and the suggested hybrid deep learning model are presented in this section. The proposed model has been tested using cutting-edge datasets, that is, NTU-RGBD, UP-FALL, UR-Fall, WISDM dataset, UCI-HAR, and the recently gathered LNMIIT-KHAD dataset (samples are shown in Fig. 9) captured by the Kinect V2 sensor. Dataset development and preprocessing are the main steps described in Section 4. After different variation in the hyperparameters, final values of the hyperparameters for the proposed model are shown in Table VI.

4.1. Model evaluation

In this section, we performed different experiments using deep learning approaches such as CNNs, LSTMs, ConvLSTM, and the suggested hybrid deep learning model. The proposed hybrid model combines the characteristics of CNNs, BiLSTM, and residual connections to efficiently capture spatial and temporal data in HAR. The architecture intends to manage sequential data and exploit local and global dependencies within input sequences. The CNN component, which is at the core of the architecture, is responsible for extracting spatial characteristics from the input skeleton data to represent joint positions over time. To downsample the feature maps and decrease spatial dimensions while preserving critical data, max-pooling layers and ReLU activation layers are added.

The sequence of joint features extracted by the CNN is then processed using the BiLSTM component. Bidirectional LSTMs enable the model to take into account both past and future information for each time step, facilitating the capture of temporal dependencies in both directions. The model's capacity to identify long-range dependencies within the sequences is improved by stacking LSTM layers. The hybrid model includes residual connections to effectively handle the difficulties of deep network training. By introducing skip connections provided by these connections, the network can learn residual functions and achieve a smoother gradient flow during training. The residual connections improve gradient propagation by reducing the degradation problem in very deep networks.

Algorithm 1. Proposed HAR algorithm.**Data:** Feature Matrices and labels**Result:** Probabilistic output per class**STEP 1** Read Kinect-V2 data.**STEP 2** Remove Null values and outliers.**STEP 3** Feature updation for selected joints.**STEP 4** Sample Equalization and Data Standardization.**while** $i \leq \text{datavalues}$ **do** **if** i is == *NULL* **then** Output \leftarrow No Activity **else** **if** i is \neq *NULL* **then** **if** $\text{data value} < \text{feature size} - \text{frame size}$ **then**

df = Features

 Features = ['Vx', 'Vy', 'Vz', 'Ax', 'Ay', 'Az', 'Kinematic Features', 'cameraX',
 'cameraY', 'cameraZ', 'OrintX', 'OrintY', 'OrintZ'] **foreach** *Fet* in *Features* **do** featureX = df[Features[Fet].values[$i : i + \text{framesize}$]] label = stats.mode(df['Activity'][$i:i+\text{framesize}$])[0][0] **end** **end** **end** **end****end****STEP 5** Split training, testing, and validation features.**STEP 6** Hyperparameter selection.**STEP 7** Model compilation and training.**STEP 8** Model Testing.

The characteristics from the CNN and BiLSTM routes are combined during the fusion and classification stage. Fully connected layers are utilized to do classification and predict the label for human activity using the fused characteristics. In order to successfully combine local and global properties, this comprehensive strategy makes use of the spatial awareness of CNNs, the sequential information handling of BiLSTMs, and the skip connections of residual connections. The proposed hybrid model has exceptional performance in HAR, correctly categorizing a wide variety of behaviors beyond the training set. It offers a potential option for HAR applications in the real world, where accurate and dependable activity recognition is crucial.

We evaluate the performance of these models on our proposed dataset and states-of-the-art datasets, that is, Wisdom Dataset [33], and UCI-HAR dataset [34], NTU- [35], UP-FALL [36], and UR-Fall [37] which are shown in Fig. 10.

A ratio of 60:20 was used to divide the dataset into train and validation sets, leaving 20% for testing. The validation dataset assessed the trained model's performance and accuracy, whereas the training set was utilized for training the classifier. Due to categorical cross-usefulness entropy for evaluating the performance of the last layer with softmax activation, the model loss is calculated using this metric. All of the models underwent 120 epochs of training. Precision, recall, F1-score, and accuracy are metrics used to assess the system's performance.

Table VI. Hyperparameter used for model training.

S. No	Parameter	Value
1	Optimizer	Adam
2	Learning rate	0.0025
3	Ragularizer l2	0.0001
4	Epochs	100
5	Batch size	128
6	Loss function	Categorical cross-entropy

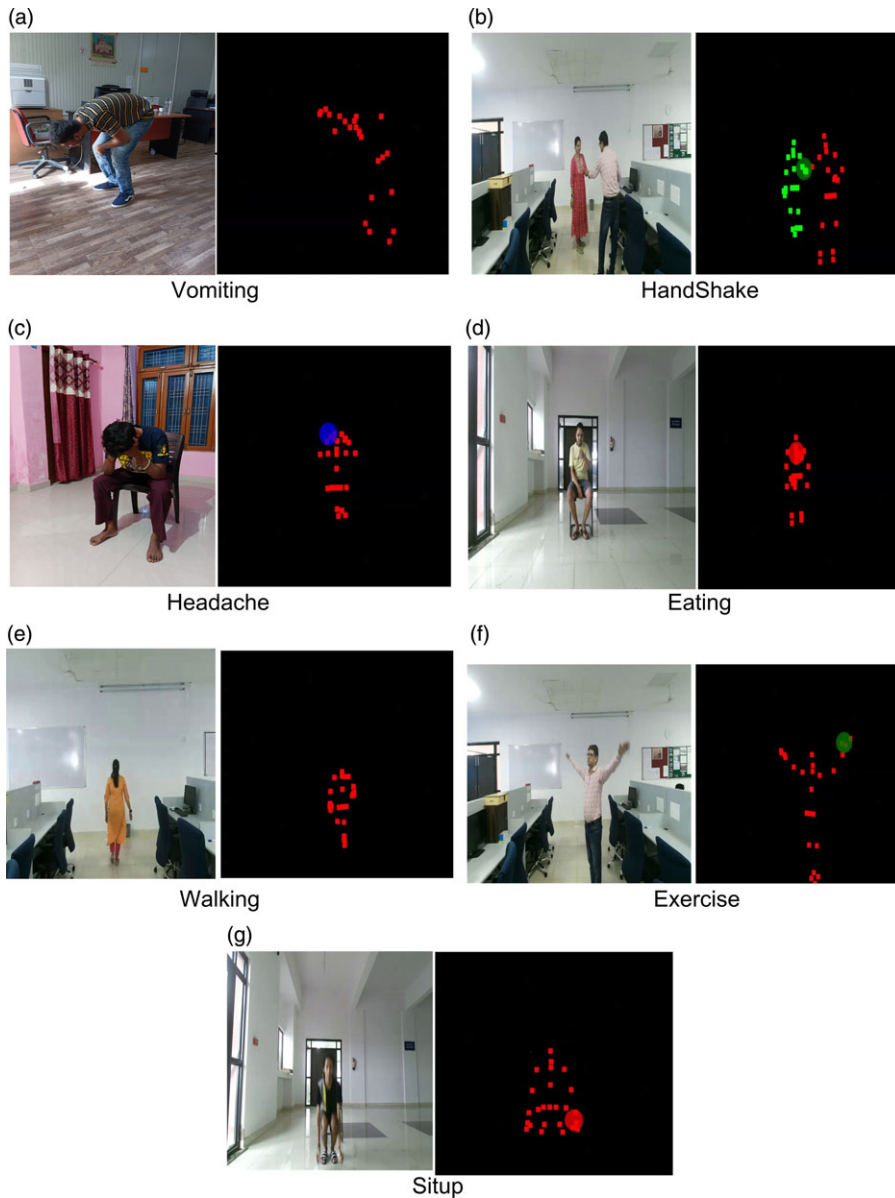


Figure 9. Human activity samples from LNMIIT-KHAD dataset.

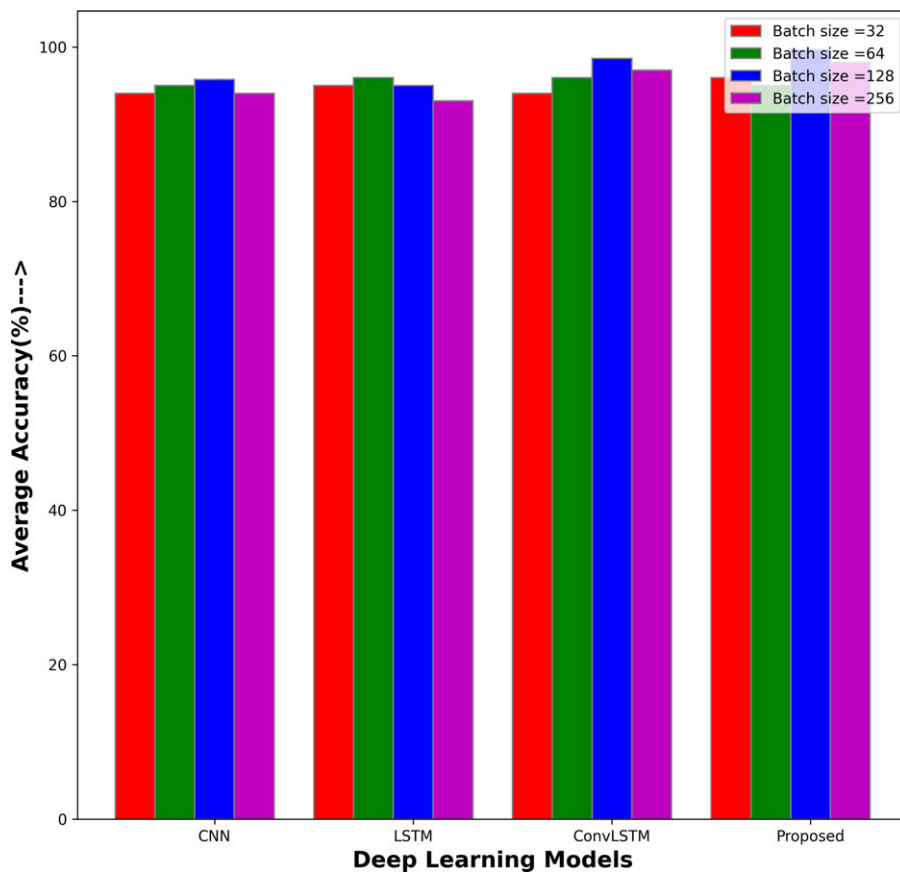


Figure 10. Model accuracy variation w.r.t batch size variation during training.

The accuracy and loss curves using LSTM, CNN, ConvLSTM, and proposed are shown in Fig. 11.

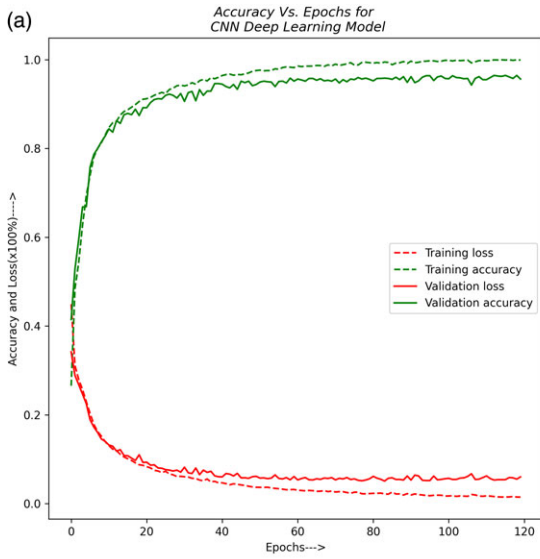
We assess the performance of the various deep learning models outlined above on our suggested dataset and two other cutting-edge datasets. When compared to other models, our proposed model had the best accuracy. Table VII displays the experimental outcomes of SOTA and proposed deep learning models. Table VIII presents the precision, recall, and F1-score values for different algorithms performed with ± 1 percentage change.

All the accuracy and plot are calculated for 120 epochs and a batch size of 128. The accuracy is also varied concerning frame size and batch size. For that, we performed various models on our proposed dataset with different batch sizes and found that the proposed model performed well for a batch size of 128 and frame size of 80, which is shown in Figs. 10 and 12.

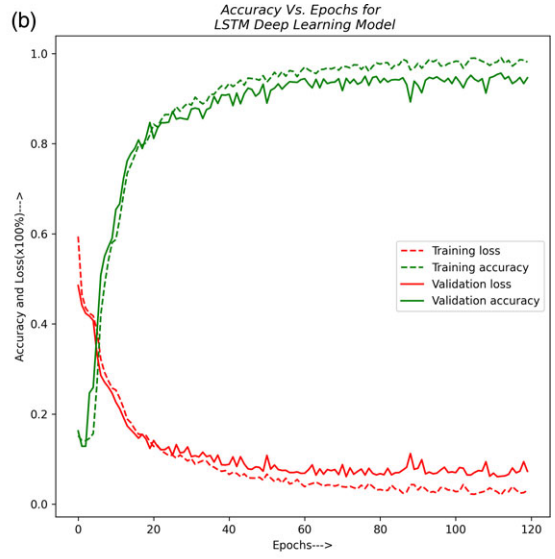
4.2. Prominent features of proposed hybrid architecture

The proposed model has been performing well on state-of-the-art datasets, proposed dataset, and also outperforms state-of-the-art deep learning models. The following are the key aspects of the proposed hybrid architecture for HAR:

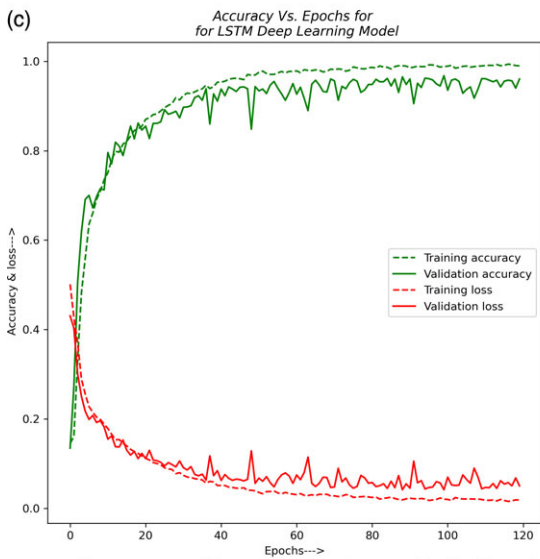
1. **Robust Handling of Illumination Variation** : The model mitigates the influence of shifting illumination conditions by effectively gathering and weighting essential joint angle information, ensuring consistent and accurate activity classification. Extensive testing on a variety of datasets



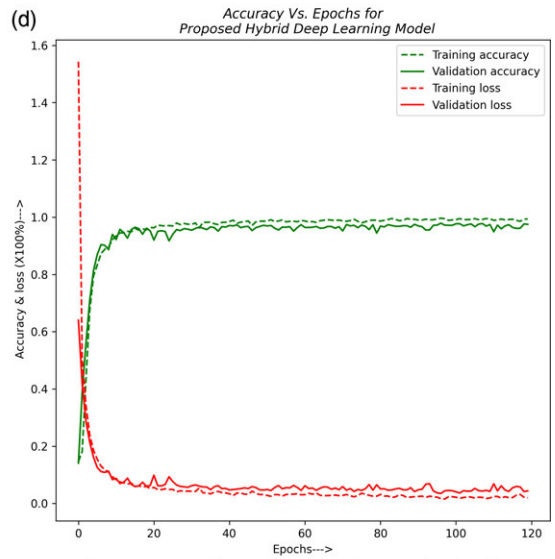
Accuracy and Loss Vs. Epochs graphs for CNN Architecture



Accuracy and Loss Vs. Epochs graphs for LSTM Architecture



Accuracy and Loss Vs. Epochs graphs for ConvLSTM Architecture



Accuracy and Loss Vs. Epochs graphs for Proposed Hybrid Deep Learning Architecture

Figure 11. Accuracy and loss versus epochs graphs for different deep learning models.

Table VII. Comparison table for different datasets for different deep learning models.

Classification models	WISDM dataset [33]		UCI-HAR dataset [34]		LNMIIT-KHAD dataset	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
CNN [22]	93.04	90.53	97.5	95	95.76	94.57
LSTM [31, 38]	94.63	91.56	92.62	90.12	97	96.80
ConvLSTM [34, 38]	97.12	95.45	93.11	92	98.89	98.32
Proposed	98.56	97.12	98.32	97.23	99.67	99.43

confirms the model's exceptional performance, demonstrating its robustness in real-world circumstances. This breakthrough holds potential for applications such as health monitoring, surveillance, and interactive systems, where lighting variance is a typical issue.

- Hand-Engineered Kinematic Features:** The model includes novel hand-engineered kinematic elements in addition to joint positioning and orientation data. These characteristics capture the kinematic qualities of human motion, improving the model's capacity to distinguish various activities.
- Cluttered Background Resilience:** The proposed model displays robustness to cluttered backgrounds without relying on depth-based data by using Kinect features and kinematic features. The Kinect features extract spatial information from skeleton joint data, allowing the model to focus on important body angles while ignoring background noise. Furthermore, the incorporation of hand-engineered kinematic features provides vital insights into motion dynamics, assisting in the differentiation of activities among clutter. This technology guarantees accurate HAR even in difficult circumstances where cluttered backdrops may interfere with conventional depth-based methods.
- Real-Time Data Performance:** The proposed architecture performs well in real-time data scenarios, suggesting its viability for practical deployment in real-world applications. Figure 13 shows the comparison between different classification models on real-time implementation.
- Temporal Information Learning:** The inclusion of bidirectional LSTM (BiLSTM) layers allows the model to capture temporal dependencies in motion sequences well. By bidirectionally analyzing joint angle data, the model acquires a thorough grasp of activity dynamics, improving its capacity to make accurate conclusions. This temporal information learning is critical for robust human activity detection, guaranteeing that the model can handle complicated and dynamic motion patterns across several activities.
- State-of-The-Art Performance:** The proposed hybrid architecture for HAR achieves state-of-the-art performance in terms of accuracy and robustness. The model effectively captures both spatial and temporal characteristics from joint motion and kinematic data by fusing CNNs and BiLSTM with residual connections. It is clearly stated from Fig. 14 percentage improvement of 2.97% in walking, 2.02% in eating, 1.85% in exercise, 3.3% in situps, and 2.94% in headache activity.
- Robust Generalization across Activity Categories:** The proposed model exhibits outstanding generalization capabilities, accurately identifying a varied range of activities that extend outside the training set. This demonstrates its outstanding adaptability and versatility to many activity categories, making it a powerful tool for real-world human activity identification applications.

Table VIII. *F1-score, recall, and precision for different deep learning models and proposed model.*

Activity name	CNN architecture			LSTM architecture			ConLSTM architecture			Proposed architecture		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Eating	0.95	0.93	0.94	0.94	0.94	0.94	0.95	0.94	0.94	0.97	0.96	0.96
Exercise	0.95	0.96	0.96	0.96	0.95	0.95	0.96	0.95	0.95	0.98	0.97	0.97
Handshake	1	0.99	0.99	1	1	1	1	0.98	0.99	1	0.99	0.99
Headache	0.93	0.95	0.94	0.93	0.94	0.93	0.94	0.96	0.95	0.97	0.97	0.97
Situps	0.94	0.93	0.94	0.95	0.94	0.94	0.95	0.95	0.95	0.98	0.97	0.97
Vomiting	0.97	0.97	0.97	0.95	0.95	0.95	0.98	0.98	0.98	0.98	0.98	0.98
Walking	0.97	0.98	0.98	0.96	0.98	0.97	0.97	0.97	0.97	0.99	0.98	0.98

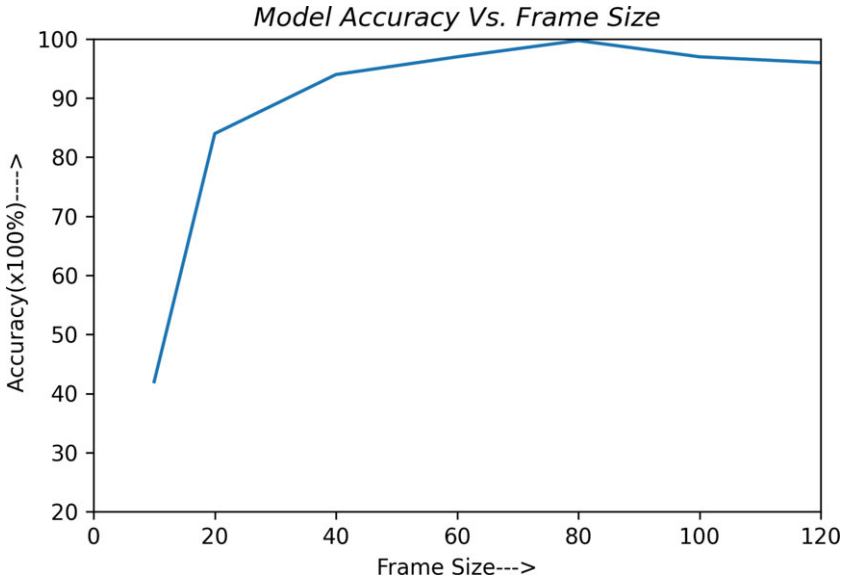


Figure 12. Model accuracy variation w.r.t feature frame size.

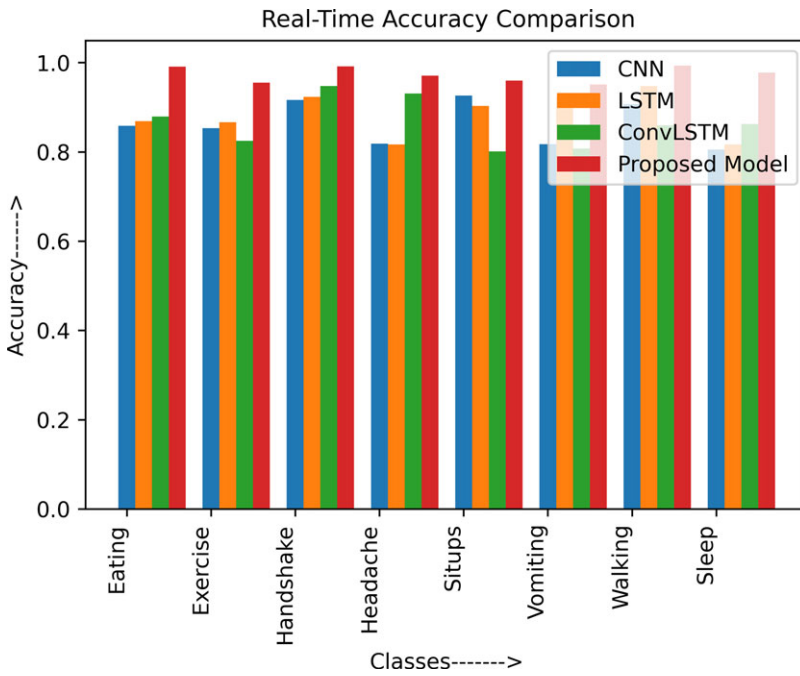


Figure 13. Real-time accuracy comparison of different classification models.

(a)

eating	95.00%	0.00%	0.00%	5.37%	0.00%	0.78%	0.38%
exercise	0.00%	95.19%	0.00%	0.00%	1.83%	0.78%	1.53%
handshake	0.00%	0.00%	100.00%	0.00%	0.37%	0.00%	0.76%
headache	4.58%	0.00%	0.00%	92.56%	0.73%	0.00%	0.00%
situps	0.00%	4.44%	0.00%	1.24%	93.77%	1.16%	0.00%
vomiting	0.42%	0.00%	0.00%	0.83%	2.20%	96.51%	0.00%
walking	0.00%	0.37%	0.00%	0.00%	1.10%	0.78%	97.33%
	eating	exercise	handshake	headache	situps	vomiting	walking

Confusion Metrics for CNN Architecture

(b)

eating	94.17%	0.00%	0.00%	4.96%	0.00%	0.78%	0.76%
exercise	0.00%	96.30%	0.00%	0.00%	1.47%	1.16%	1.53%
handshake	0.00%	0.00%	100.00%	0.00%	0.37%	0.39%	1.15%
headache	5.00%	0.00%	0.00%	92.98%	0.73%	0.00%	0.38%
situps	0.00%	3.33%	0.00%	1.24%	94.87%	1.16%	0.00%
vomiting	0.83%	0.00%	0.00%	0.83%	1.10%	95.35%	0.00%
walking	0.00%	0.37%	0.00%	0.00%	1.47%	1.16%	96.18%
	eating	exercise	handshake	headache	situps	vomiting	walking

Confusion Metrics for LSTM Architecture

(c)

eating	95.00%	0.00%	0.00%	4.13%	0.00%	0.39%	0.76%
exercise	0.00%	96.30%	0.00%	0.00%	1.10%	0.78%	1.15%
handshake	0.00%	0.00%	100.00%	0.00%	0.37%	0.00%	0.76%
headache	4.58%	0.00%	0.00%	94.21%	1.47%	0.00%	0.38%
situps	0.00%	2.59%	0.00%	0.83%	94.87%	0.39%	0.00%
vomiting	0.42%	0.00%	0.00%	0.83%	1.47%	98.06%	0.00%
walking	0.00%	1.11%	0.00%	0.00%	0.73%	0.39%	96.95%
	eating	exercise	handshake	headache	situps	vomiting	walking

Confusion Metrics for ConvLSTM Architecture

(d)

eating	97.08%	0.00%	0.00%	2.49%	0.00%	0.39%	0.00%
exercise	0.00%	98.15%	0.00%	0.00%	0.00%	0.78%	0.38%
handshake	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%
headache	2.92%	0.00%	0.00%	97.51%	0.37%	0.00%	0.00%
situps	0.00%	1.48%	0.00%	0.00%	98.17%	0.39%	0.00%
vomiting	0.00%	0.00%	0.00%	0.00%	1.10%	98.06%	0.00%
walking	0.00%	0.37%	0.00%	0.00%	0.37%	0.39%	99.62%
	eating	exercise	handshake	headache	situps	vomiting	walking

Confusion Metrics for Proposed Hybrid Deep Learning Architecture

Figure 14. Confusion matrices for proposed and state-of-the-art architectures.

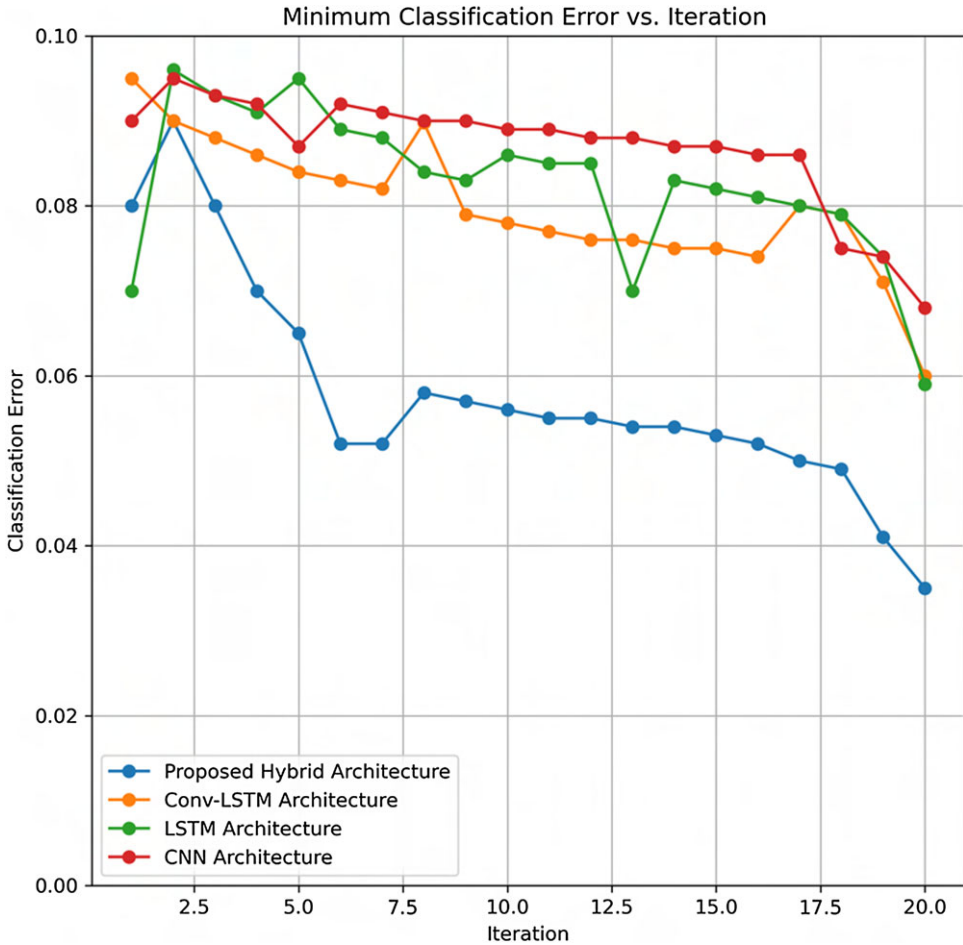


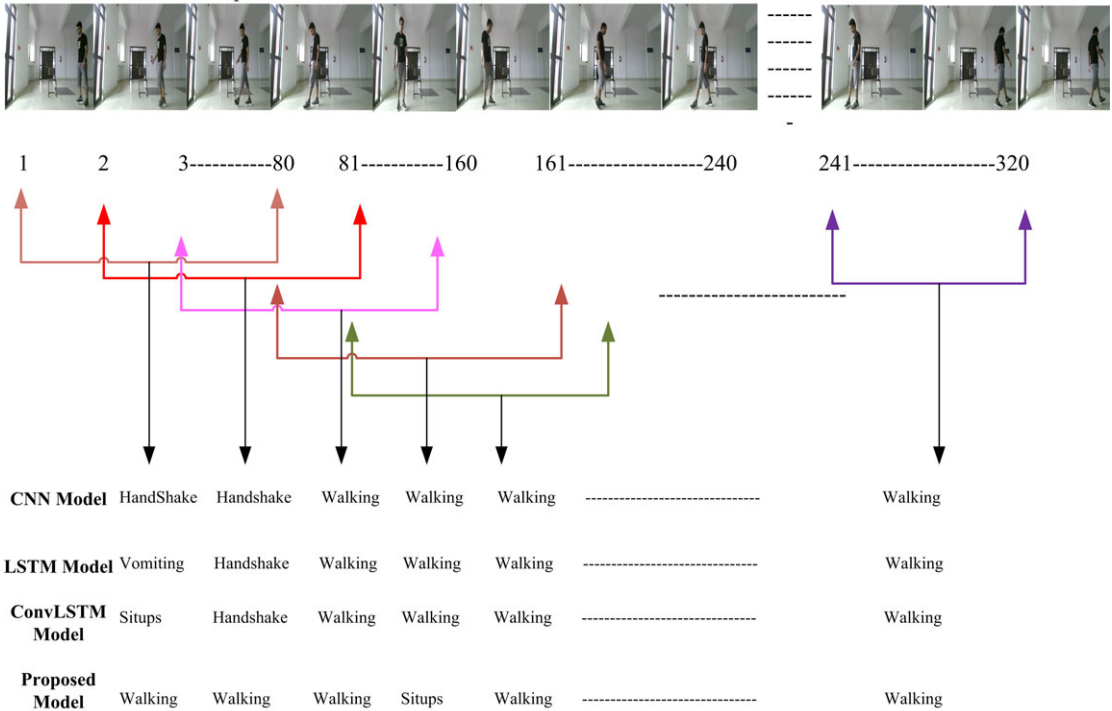
Figure 15. Classification error for different deep learning models.

As the proposed model has a fusion of the CNN and BiLSTM layers, BiLSTM networks process input sequences simultaneously in the forward and backward directions. This enables the network to gather contextual data from both previous and upcoming time steps. It facilitates a deeper comprehension of the sequence’s components’ connections and dependencies and handles long-term dependencies. Due to that, the proposed model has less misclassification error as a comparison to state-of-the-art models, as shown in Fig. 15.

In the proposed methodology, the complete video has been converted into video frames. Trained model has been predicted the activity based on the video frame set of 80. Figures 16 and 17 show the prediction accuracies for walking and vomiting activity. The accuracy of our proposed model outperforms existing state-of-the-art deep learning models. Precision, recall, and F1-score are also better than state-of-the-art deep learning models, which is shown in Tables VIII and IX. The Precision, recall, and F1-score curve for LSTM, CNN, convLSTM, and the proposed approach are shown in Fig. 18.

Let's break down the calculation:

1. Each set contains 80 frames.
2. The video has 320 frames.
3. During prediction, we use a sliding window approach with a stride of 1 frame, meaning we shift one frame at a time for the next prediction.



4. Calculate the total number of sets that can be made from the video.

$$\text{Total sets} = (\text{Number of frames in video}) - (\text{Size of each set}) + 1$$

$$\text{Total sets} = 320 - 80 + 1 = 241$$

#1 CNN Model Accuracy :	Number of correctly predicted sets = 229	Accuracy = 95%
#2 LSTM Model Accuracy:	Number of correctly predicted sets = 232	Accuracy = 96.26%
#3 Conv-LSTM Model Accuracy:	Number of correctly predicted sets = 232	Accuracy = 96.26%
#4 Proposed Model Accuracy:	Number of correctly predicted sets = 235	Accuracy = 97.51 %

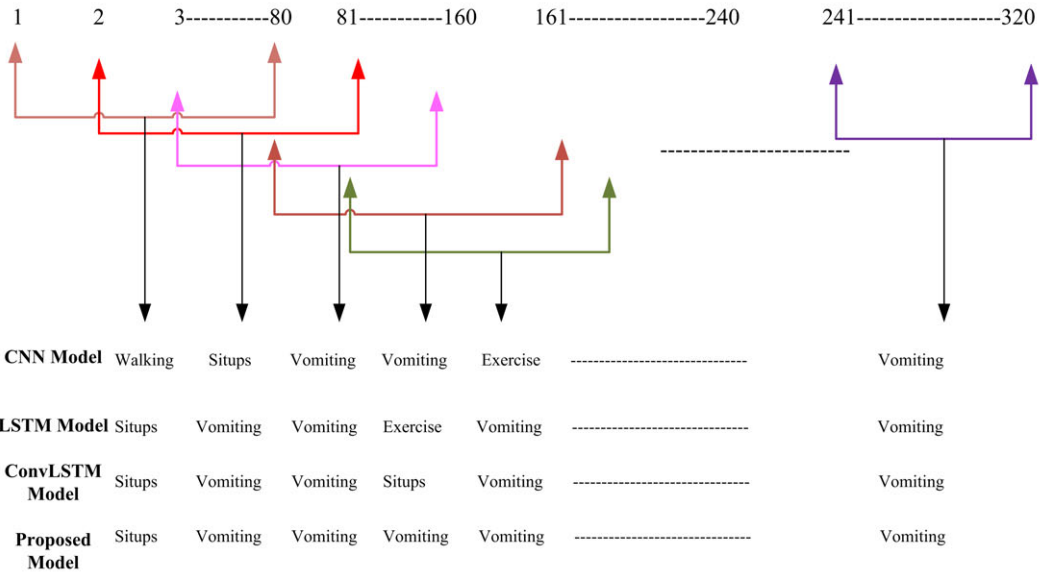
5. Total number of videos in Walking Activity class: 150

#1 CNN Model Accuracy :	Number of correctly predicted videos ≈ 146	Accuracy = 97.33%
#2 LSTM Model Accuracy:	Number of correctly predicted videos ≈ 144	Accuracy = 96%
#3 Conv-LSTM Model Accuracy:	Number of correctly predicted videos ≈ 145	Accuracy = 96.66%
#4 Proposed Model Accuracy:	Number of correctly predicted videos ≈ 149	Accuracy = 97.51 %

Figure 16. Accuracy calculation for walking activity using different classification models.

Let's break down the calculation:

1. Each set contains 80 frames.
2. The video has 289 frames. Padded 320-289 = 31 frames in the last. Which are the copy of last 31 frames.
3. During prediction, we use a sliding window approach with a stride of 1 frame, meaning we shift one frame at a time for the next prediction.



4. Calculate the total number of sets that can be made from the video.

Total sets = (Number of frames in video) - (Size of each set) + 1
 Total sets = 320 - 80 + 1 = 241

#1 CNN Model Accuracy :	Number of correctly predicted sets = 231	Accuracy = 96.11%
#2 LSTM Model Accuracy:	Number of correctly predicted sets = 229	Accuracy = 95.02%
#3 Conv-LSTM Model Accuracy:	Number of correctly predicted sets = 234	Accuracy = 97.10%
#4 Proposed Model Accuracy:	Number of correctly predicted sets = 237	Accuracy = 98.34 %

5. Total number of videos in Vomiting Activity class: 150

#1 CNN Model Accuracy :	Number of correctly predicted videos ≈ 145	Accuracy = 97.33%
#2 LSTM Model Accuracy:	Number of correctly predicted videos ≈ 144	Accuracy = 96%
#3 Conv-LSTM Model Accuracy:	Number of correctly predicted videos ≈ 147	Accuracy = 98%
#4 Proposed Model Accuracy:	Number of correctly predicted videos ≈ 147	Accuracy = 98 %

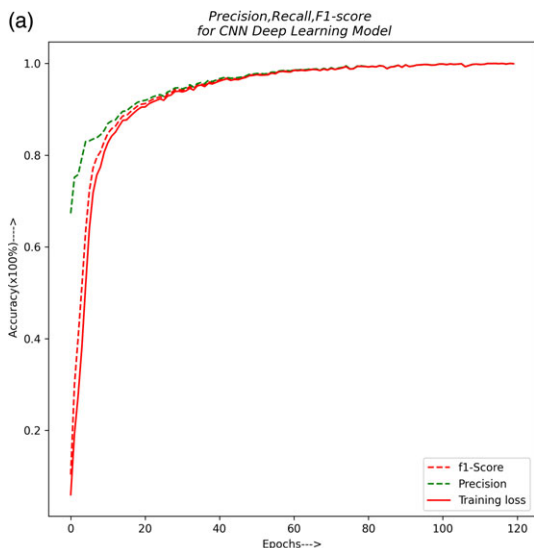
Figure 17. Accuracy calculation for vomiting activity using different classification models.

Table IX. Performance comparison table for LNMIIT-KHAD and NTU-RGBD dataset.

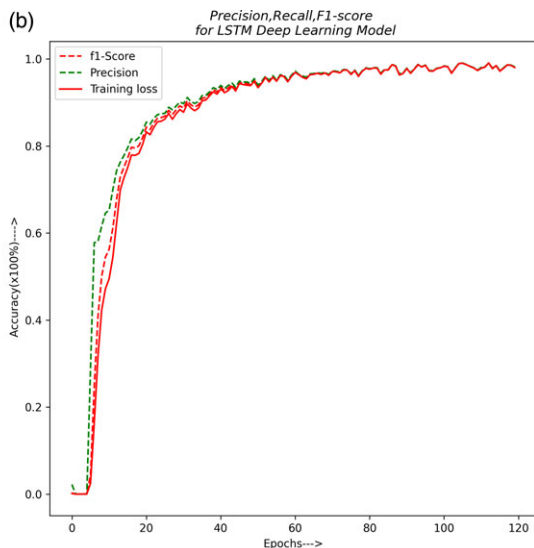
Classification models	Dataset	Testing accuracy (%)	mAP (%)	F1-score (%)
CNN [22]	LNMIIT-KHAD	94.20	93.10	92.89
	NTU RGBD	94.75	92.45	92.40
LSTM [31, 38]	LNMIIT-KHAD	92.62	92.15	92.05
	NTU RGBD	93.90	92.50	92.15
ConvLSTM [38, 39]	LNMIIT-KHAD	93.11	92.56	92.50
	NTU RGBD	92.05	91.10	990.95
GRU-INC [40]	LNMIIT-KHAD	95.85	92.40	92.40
	NTU RGBD	94.12	91.15	91.00
Inception CNN-GRU [40]	LNMIIT-KHAD	96.45	92.60	92.00
	NTU RGBD	93.45	91.02	90.80
Deep RNN [41]	LNMIIT-KHAD	92.15	89.55	89.10
	NTU RGBD	93.50	90.05	90.00
AFCDL [4]	LNMIIT-KHAD	98.12	97.50	97.25
	NTU RGBD	97.59	96.10	96.00
Proposed (ours)	LNMIIT-KHAD	98.32	97.60	97.515
	NTU RGBD	96.52	94.60	94.25

5. Conclusion and future scope

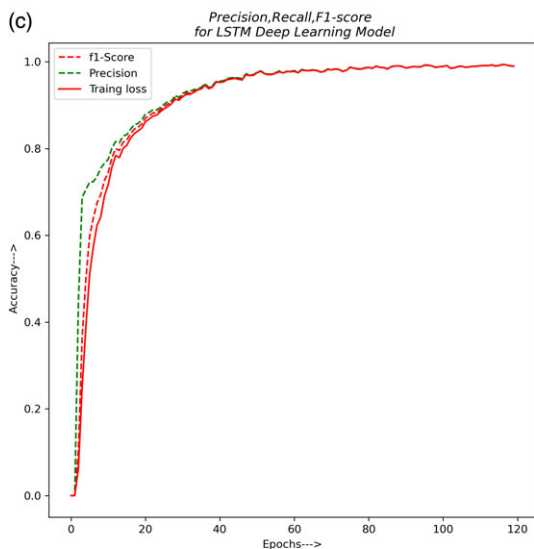
A hybrid deep learning model and a single Kinect V2 sensor have been used as an activity identification system that protects user privacy. Primary skeleton coordinates and geometrical and kinematic information are inputted into the proposed deep learning network. The user's privacy is safeguarded since the system only uses derived features and basic skeleton joint coordinates, not the user's real photographs. On the LNMIIT-KHAD dataset and leading datasets, the performance of the deep learning-based classification algorithms such as CNN, LSTM, ConvLSTM, and the suggested model has been compared. The recommended approach correctly identifies human behaviors, including eating, exercising, situps, headache, vomiting, shaking hands, and walking. The proposed model's accuracy of 99.5% surpasses that of CNN, LSTM, and ConvLSTM, which have accuracy rates of 95.76%, 97%, and 98.89%, respectively. To evaluate the performance of the proposed model, it has been tested on other additional datasets, that is, NTU-RGBD [35], UP-FALL [36], and UR-Fall [37]. The testing accuracies are shown in Table X. The suggested method has been tested in real time and discovered to be independent of the stance, individual, clothes, etc. The dataset sample is accessible to the general public. In the future, we want to include more complicated physical activities and develop a model that can detect the activity of several people at once. We will also investigate advanced deep learning-based approaches such as reinforcement, lifetime, incremental, and active learning for activity recognition. We also have plans to create a sizable HAR dataset, including a variety of daily activities and physical activities.



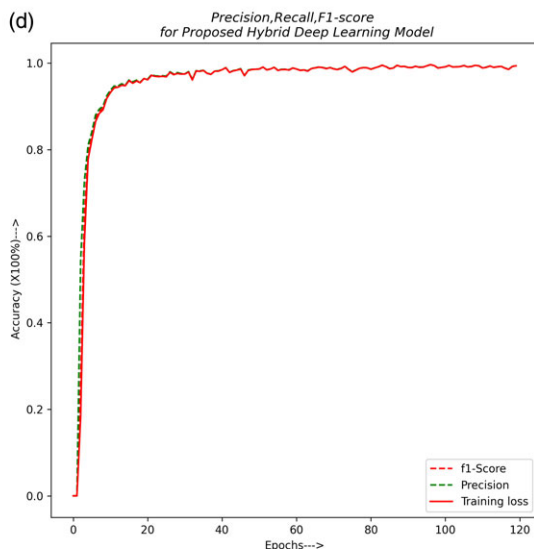
Precision, Recall and F1-score variation for CNN Architecture



Precision, Recall and F1-score variation for LSTM Architecture



Precision, Recall and F1-score variation for ConvLSTM Architecture



Precision, Recall and F1-score variation for Proposed Hybrid Deep learning Architecture

Figure 18. Precision, recall, and F1-score variation for different deep learning models.

Table X. Model testing accuracy on different SOTA datasets and proposed dataset.

Classification methods	SOTA datasets “Testing Accuracy”			
	NTU-RGBD [35]	UP-FALL [36]	UR-Fall [37]	LNMIIT-KHAD dataset
CNN [22]	Eat meal = 88% Vomiting = 90.12% Headache = 91.04% Shaking Hands = 93.45% Walking = 94.75%	Walking = 90% Standing = 94.40% Sitting = 95.14% eating an apple = 92% Falling Forward = 93.12%	Walking = 89.10% Eating = 85.65% Falling Forward = 90% Drinking Water = 90% Standing = 95% Sitting = 94.12%	Eating = 95% Exercise = 95.19% Handshake = 100% Headache = 92.56% Situps = 93.77% Vomiting = 96.51% Walking = 97.33%
LSTM [31, 38]	Eat meal = 92.34% Vomiting = 90.56% Headache = 93% Shaking Hands = 93.12% Walking = 94.90%	Walking = 89.45% Standing = 95.25% Sitting = 98% eating an apple = 92.14% Falling Forward = 93%	Walking = 90.30% Eating = 88.10% Falling Forward = 91% Drinking Water = 93% Standing = 95.10% Sitting = 95%	Eating = 94.17% Exercise = 96.30% Handshake = 100% Headache = 92.98% Situps = 94.87% Vomiting = 95.35% Walking = 96.18%
ConvLSTM [34, 38]	Eat meal = 91.25% Vomiting = 93.50% Headache = 92.05% Shaking Hands = 93% Walking = 96.78%	Walking = 92.78% Standing = 95.45% Sitting = 98.12% eating an apple = 93.34% Falling Forward = 96.80%	Walking = 90.48% Eating = 90.25% Falling Forward = 92.10% Drinking Water = 92.16% Standing = 96.45% Sitting = 96.20%	Eating = 95% Exercise = 96.30% Handshake = 100% Headache = 94.21% Situps = 94.87% Vomiting = 98.06% Walking = 96.95%
Proposed-ours	Eat meal = 94.45% Vomiting = 95.32% Headache = 96.75% Shaking Hands = 96.52% Walking = 97.80%	Walking = 95.90% Standing = 96.35% Sitting = 98.50% eating an apple = 97% Falling Forward = 98.12%	Walking = 93% Eating = 92.10% Falling Forward = 92% Drinking Water = 94% Standing = 97.12% Sitting = 97.80%	Eating = 97.08% Exercise = 98.15% Handshake = 100% Headache = 97.51% Situps = 98.17% Vomiting = 98.03% Walking = 99.62%

Author contributions. Rabul Laskar, Joyeeta Singha, and Sandeep Saini conceived and designed the study. Manoj kumar sain conducted data gathering, performed model analyses, and wrote the article.

Financial support. The paper is supported by DST (Govt. of India) under the SEED Division [SP/YO/407/2018]. Dr. Joyeeta Singha is the principal investigator.

Competing interests. The authors declare no Competing interests exist.

Ethical approval. Not applicable.

Dataset availability. The author declares that the dataset will be available on request. Please contact the corresponding author.

Declaration. The author declares that the manuscript is prepared as per the journal's guidelines for authors.

References

- [1] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi and J. S. Suri, "Human activity recognition in artificial intelligence framework: A narrative review," *Artif. Intell. Rev.* **55**(6), 4755–4808 (2022).
- [2] S. Zhang, 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 (2022).
- [3] Y. Zhang, F. Zhang, Y. Jin, Y. Cen, V. Voronin and S. Wan, "Local correlation ensemble with gcn based on attention features for cross-domain person re-id," *ACM Trans. Multimed. Comput. Commun. Appl.* **19**(2), 1–22 (2023).
- [4] Z. Gao, H. Z. Xuan, H. Zhang, S. Wan and K. K. R. Choo, "Adaptive fusion and category-level dictionary learning model for multiview human action recognition," *IEEE Internet Things J.* **6**(6), 9280–9293 (2019).
- [5] N. Nair, C. Thomas and D. B. Jayagopi, "Human Activity Recognition Using Temporal Convolutional Network," In: *iWOAR'18: Proceedings of the 5th International Workshop on Sensor-based Activity Recognition and Interaction* (Association for Computing Machinery, 2018).
- [6] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu and J. Zhang, "Convolutional Neural Networks for Human Activity Recognition Using Mobile Sensors," In: *6th International Conference on Mobile Computing, Applications and Services* (2014) pp. 197–205.
- [7] S. Dhanraj, S. De and D. Dash, "Efficient Smartphone-based Human Activity Recognition Using Convolutional Neural Network," In: *2019 International Conference on Information Technology (ICIT)* (2019) pp. 307–312.
- [8] J. Wang, Y. Chen, S. Hao, X. Peng and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recogn. Lett.* **119**(1), 3–11 (2019). <https://www.sciencedirect.com/science/article/pii/S016786551830045X>
- [9] Q. Ke, M. Bennamoun, S. An, F. Sohel and F. Boussaid, "A New Representation of Skeleton Sequences for 3D Action Recognition," In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). <http://dx.doi.org/10.1109/CVPR.2017.486>
- [10] H. Liu, J. Tu and M. Liu, Two-stream 3D convolutional neural network for skeleton-based action recognition (2017). ArXiv, abs/1705.08106
- [11] W. Ding, K. Liu, E. Belyaev and F. Cheng, "Tensor-based linear dynamical systems for action recognition from 3D skeletons," *Pattern Recogn.* **77**(1), 75–86 (2018). <https://www.sciencedirect.com/science/article/pii/S0031320317304909>
- [12] A. Scano, A. Chiavenna, M. Malosio, L. M. Tosatti and F. Molteni, "Kinect v2 implementation and testing of the reaching performance scale for motor evaluation of patients with neurological impairment," *Med. Eng. Phys.* **56**(1), 54–58 (2018). <https://www.sciencedirect.com/science/article/pii/S1350453318300596>
- [13] V. Bijalwan, V. B. Semwal and V. Gupta, "Wearable sensor-based pattern mining for human activity recognition: Deep learning approach," *Ind. Robot* **49**(1), 21–33 (2022).
- [14] R. Jain, V. B. Semwal and P. Kaushik, "Deep ensemble learning approach for lower extremity activities recognition using wearable sensors," *Expert Syst.* **39**(6), e12743 (2022).
- [15] V. Bijalwan, V. B. Semwal, G. Singh and T. K. Mandal, "HDL-PSR: Modelling spatio-temporal features using hybrid deep learning approach for post-stroke rehabilitation," *Neural Process. Lett.* **55**(1), 279–298 (2023).
- [16] V. B. Semwal, A. Gupta and P. Lalwani, "An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition," *J Supercomput.* **77**(11), 12256–12279 (2021).
- [17] 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 (2021).
- [18] S. K. Yadav, K. Tiwari, H. M. Pandey and S. A. Akbar, "Skeleton-based human activity recognition using convlstm and guided feature learning," *Soft Comput.* **26**(2), 877–890 (2022).
- [19] K. Ashwini, R. Amutha and S. A. raj, "Skeletal Data Based Activity Recognition System," In: *2020 International Conference on Communication and Signal Processing (ICCSPP)* (2020) pp. 444–447.
- [20] J. Liu, G. Wang, P. Hu, L.-Y. Duan and A. C. Kot, "Global Context-Aware Attention LSTM Networks for 3D Action Recognition," In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017) pp. 3671–3680.

- [21] Y. Yan, T. Liao, J. Zhao, J. Wang, L. Ma, W. Lv, J. Xiong and L. Wang, Deep transfer learning with graph neural network for sensor-based human activity recognition (2022). <http://arxiv.org/abs/2203.07910>
- [22] X. Jiang, Y. Lu, Z. Lu and H. Zhou, *Smartphone-Based Human Activity Recognition Using CNN in Frequency Domain: APWeb-WAIM 2018 International Workshops: MWDA, BAH, KGMA, DMMOOC, DS, Macau, China, July 23-25, 2018, Revised Selected Papers* (2018) pp. 101–110.
- [23] M. Gholamrezai and S. M. T. Almodarresi, “Human Activity Recognition Using 2D Convolutional Neural Networks,” In: *2019 27th Iranian Conference on Electrical Engineering (ICEE)* (2019) pp. 1682–1686.
- [24] A. Abedin, E. Abbasnejad, Q. Shi, D. Ranasinghe and H. Rezatofghi, “Deep Auto-Set: A Deep Auto-Encoder-Set Network for Activity Recognition Using Wearables,” In: *MobiQuitous '18: Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services* (2018) pp. 246–253.
- [25] S. Yu and L. Qin, “Human Activity Recognition with Smartphone Inertial Sensors Using Bidir-LSTM Networks,” In: *2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)* (2018) pp. 219–224.
- [26] Y. Zhao, R. Yang, G. Chevalier, X. Xu and Z. Zhang, “Deep residual Bidir-LSTM for human activity recognition using wearable sensors,” *Math. Probl. Eng.* **2018**, 1–13 (2018).
- [27] S. Deep and X. Zheng, “Hybrid Model Featuring CNN and LSTM Architecture for Human Activity Recognition on Smartphone Sensor Data,” In: *2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)* (2019) pp. 259–264.
- [28] K. Xia, J. Huang and H. Wang, “Lstm-cnn architecture for human activity recognition,” *IEEE Access* **8**(1), 56855–56866 (2020).
- [29] H. Chen, G. Wang, J.-H. Xue and L. He, “A novel hierarchical framework for human action recognition,” *Pattern Recognit.* **55**(1), 148–159 (2016).
- [30] O. Banos, J. M. Galvez, M. Damas, H. Pomares and I. Rojas, “Window size impact in human activity recognition,” *Sensors (Switzerland)* **14**(4), 6474–6499 (2014).
- [31] S. Mekruksavanich and A. Jitpattanukul, “LSTM networks using smartphone data for sensor-based human activity recognition in smart homes,” *Sensors* **21**(5), 1–25 (2021).
- [32] I. U. Khan, S. Afzal and J. W. Lee, “Human activity recognition via hybrid deep learning based model,” *Sensors* **22**(1), 323 (2022).
- [33] H. Salim, M. Alaziz and T. Abdalla, “Human activity recognition using the human skeleton provided by kinect,” *Iraqi J. Electr. Electron. Eng.* **17**(2), 183–189 (2021).
- [34] M. A. Khatun, M. A. Yousuf, S. Ahmed, M. Z. Uddin, S. A. Alyami, S. Al-Ashhab, H. F. Akhdar, A. Khan, A. Azad and M. A. Moni, “Deep CNN-LSTM with self-attention model for human activity recognition using wearable sensor,” *IEEE J. Transl. Eng. Health Med.* **10**(1), 1–16 (2022).
- [35] A. Shahroudy, J. Liu, T.-T. Ng and G. Wang, “Ntu rgb+d: A large scale dataset for 3D human activity analysis,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016) pp. 1010–1019.
- [36] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez and C. Peñafort-Asturiano, “Up-fall detection dataset: A multimodal approach,” *Sensors (Switzerland)* **19**(9), 1988 (2019).
- [37] A. Lotfi, S. Albawendi, H. Powell, K. Appiah and C. Langensiepen, “Supporting independent living for older adults; employing a visual based fall detection through analysing the motion and shape of the human body,” *IEEE Access* **6**(1), 70272–70282 (2018).
- [38] N. Varshney, B. Bakariya, A. K. S. Kushwaha and M. Khare, “Human activity recognition by combining external features with accelerometer sensor data using deep learning network model,” *Multimed. Tools Appl.* **81**(24), 34633–34652 (2022). doi: [10.1007/s11042-021-11313-0](https://doi.org/10.1007/s11042-021-11313-0).
- [39] T. R. Mim, M. Amatullah, S. Afreen, M. A. Yousuf, S. Uddin, S. A. Alyami, K. F. Hasan and M. A. Moni, “Gru-inc: An inception-attention based approach using gru for human activity recognition,” *Expert Syst. Appl.* **216**(2), 119419 (2023).
- [40] N. Dua, S. N. Singh, V. B. Semwal and S. K. Challa, “Inception inspired CNN-GRU hybrid network for human activity recognition,” *Multimed. Tools Appl.* **82**(4), 5369–5403 (2023).
- [41] A. Usmani, N. Siddiqui and S. Islam, “Skeleton joint trajectories based human activity recognition using deep RNN,” *Multimed. Tools Appl.* (2023). doi: [10.1007/s11042-023-15024-6](https://doi.org/10.1007/s11042-023-15024-6).