An Adaptive Batch Size-Based-CNN-LSTM Framework for Human Activity Recognition in Uncontrolled Environment

Nurul Amin Choudhury and Badal Soni

***Abstract*—Human activity recognition (HAR) is a process of identifying the daily living activities of an individual using a set of sensors and appropriate learning algorithms. Most of the works on HAR are done using a mix of sensor data that is collected in a simulated environment, and due to that, the real-time recognition suffers. This article proposes an efficient adaptive batch size-based-CNN-LSTM model for recognizing different human activities in an uncontrolled environment. It uses adaptive batch sizes from 128 to 1024 for iterative model training and validation. The proposed model can handle imbalanced classes and un-normalized data efficiently. A state-of-art HAR dataset is also generated in an open environment to get the activity data of an uncontrolled scenario. With minimal data preprocessing and data augmentation, the model is tested, and the proposed model managed to get the highest accuracy of 99.29% with an average loss of 0.08 $\pm$ 0.136%. The presented model is also tested with two public datasets named- mHealth and MotionSense and achieved an accuracy of 99.5% and 99.8%, respectively. The proposed model outperforms all the previous benchmarks and approaches by a good margin.**

***Index Terms*—Activities of daily living (ADL), convolutional neural network (CNN), human activity recognition (HAR), long short term memory (LSTM).**

I. Introduction

IN RECENT years, wearable sensors have seen significant manufacturing and performance advancements. The application of wearable sensors is numerous in health care, home automation, industries, and many more. One of the most explored fields using wearable sensors is human activity recognition (HAR) [1], [2], [3], [4]. Any movement produced by human muscles and bones by exerting a small amount of energy is known as human activity. The process of identifying those human activities with the help of a mix of sensors, learning algorithms, and appropriate preprocessing techniques is termed as HAR. HAR is a complex process as there is no clear way to commune the collected raw sensor data to different human activities directly. The raw data and activity labels often differ, causing the classifiers not to learn optimally. Most researchers [3], [4] collect the data in the simulated environment and train and test their models on those data. Due to this, the real-time performance degrades, and the online activity recognition suffers from errors. The data collected in a simulated environment like a laboratory does not make subjects restful, leading to poor data collection.

Deep learning algorithms are prevalent among researchers [5], [6] because of their performance and model generalization capabilities. The use of different deep learning algorithms varies from a simple task of email spam detection to a complex task of identifying cancer and human activities. From the ability to automatically extract and select the features to the flexible architecture for dynamic adaptation, deep learning successfully handles a vast range of application domains. The primary need for getting the optimal results is to provide sufficient data for the models because of its data-hungry nature. The dynamic nature of deep learning models is handled by the adaptive weights that can be adjusted by an iterative model training and rigorous model validation.

The use of smartphone users is also increasing year by year and most of the population nowadays has smartphones with them, it will be one of the most suitable and realistic data collection modules. The data can be collected while doing all the ADLs without disturbing the subject and in a natural way. Along with this, as the trend of smartwatches is also popular, and people are frequently using them for tracking and monitoring services, we could easily use them for ADLs data collection as well.

Research enthusiast exploits the field of HAR [2], [7] in different forms in terms of the type of data acquisition module and learning algorithms. Based on the data acquisition module, the HAR is classified into various types: extrinsic sensor-based, wearable sensor-based, vision-based, and hybrid approaches. One of the most exploited approaches is the wearable sensor-based HAR system. Here, wearable sensors like gyroscopes, accelerometers, pedometers, etc., are used for precise data collection. A suitable mounting location is also defined for getting optimal raw data, and with the help of various learning algorithms, different human activities are recognized.

In this article, we have proposed a novel CNN-LSTM model for sensor-based HAR using an adaptive batch size framework in an uncontrolled environment. Most of the ADLs data are collected in a human-controlled environment for developing

Manuscript received 18 May 2022; revised 17 October 2022; accepted 9 December 2022. Date of publication 30 January 2023; date of current version 11 August 2023. Paper no. TII-22-2132. *(Corresponding author: Nurul Amin Choudhury.)*

The authors are with the Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar 788010, India (e-mail: nurul0400@gmail.com; badal@cse.nits.ac.in).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3229522.

Digital Object Identifier 10.1109/TII.2022.3229522

HAR systems. This degrades the real-time activity recognition results as the environmental factor mismatches with the human-controlled environment. We created our own sensor-based HAR dataset using smartphone accelerometers and gyroscope sensors in an uncontrolled environment. It helps mitigate real-life environmental factors, and real-time recognition does not get hindered by domain shifts. Then, we segregated different data instances based on a sample window protocol. Later, we developed a hybrid CNN-LSTM model that trains the sensor data using adaptive batch sizes in a single training cycle. This ensures low time complexity and better model generalization as the model learns gradually from the transferred data instances.

The main contribution and novelties of this work are as follows.

1) A novel adaptive batch-size-based-CNN-LSTM model is proposed for a smartphone sensor-based HAR system in an uncontrolled environmental scenario that improves overall model generalization and computational efficiency.
2) A time-interval-based-sliding window selection scheme for segment reshaping is proposed and described for selecting the prominent data instances from different time windows.
3) Various deep learning models and two state-of-art public HAR datasets, namely, mHealth and MotionSense are exploited, and the performance is compared with the proposed model in terms of most of the incorporated evaluation metrics.
4) A state-of-art HAR dataset is generated with the help of inbuilt smartphone sensors - accelerometer and gyroscope in an uncontrolled environment to get the real raw data of different human activities.

The rest of this article is organized as follows. Section II summarizes the recent state-of-art works on sensor-based HAR systems with publicly available datasets. Section III describes the proposed model and incorporated evaluation matrices and Section IV portrays the experimental results and discussion on the achieved results. Finally, Section V concludes this article.

II. Related Works

Researchers [1], [2] have used shallow, ensemble, and deep learning algorithms for performing human activity recognition. With good quality data, even shallow and ensemble [4] learning yields promising results, but it needs domain expertise and intense data preprocessing. The manual necessity of feature engineering and intense data preprocessing forces the researchers to move toward deep learning models [3], [8] to conceive automatic feature engineering and robust results. The latest trend in deep learning-based HAR is outlined as follows.

Mutegeki et al. [9] proposed a CNN-LSTM approach for HAR. They employed two publicly available datasets named iSPL and UCI-HAR [10] and used 1-D-CNN for feature extraction and data preprocessing. Later they applied LSTM with softmax activation for activity recognition. Upon comparing all the employed models, they got the highest accuracy of 99% with the iSPL dataset and 92.13% with the UCI-HAR dataset. In [3], the authors tested different LSTM networks for sensor-based HAR. They also used the UCI-HAR [10] dataset and did two forms of the sampling window generation process. Testing and cross-validating various networks, they discovered that 4-layer CNN-LSTM yields the most optimal results. Mohsen et al. [11] presented three models for testing the HAR system using a valuable hyperparameter technique for obtaining efficient performance. They used CNN-LSTM as LSTM can efficiently handle the temporal data, and CNN can remarkably manage spatial data. Also, the complex feature system is smoothly tackled by the CNN architecture, and upon testing their models, they achieve the best accuracy with the hybrid CNN-LSTM model.

Singh et al. [12] proposed a deep ConvLSTM with self-attention module for a sensor-based HAR system. They focused on self-attention mechanism that learns and discover the relationship between time stamps in the input vector. Incorporating softmax as a classification activation function, they managed to achieve the highest accuracy of 97.65%. In [13], the authors proposed HAR on limited sensory data using various deep learning algorithms. Their dataset suffers from a class imbalance problem, which they tackle with the help of a data augmentation cum class balancing algorithm. Mekruksavanich et al. [14] used a two-layer hybrid CNN-LSTM to classify human activities collected from the smartwatch's inbuilt sensors. Testing their model with the integration of Bayesian optimization, they managed to achieve an average accuracy of 96.2%. With the increase in activity complexity with the UCI-HAR dataset, their approach fails to produce comparable results.

Mukherjee et al. [22] proposed a new ensemble deep learning method named *EnsemConvNet* for smartphone sensor-based HAR for health care applications. They combined three classification standards—Encoded-Net, CNN-Net, and CNN-LSTM and named it *EnsemConvNet*. The data are handed to all the models, and the result is envisioned using maximal voting architecture. Challa et al. [23] also envisioned an ensemble-based multi-branch CNN-biLSTM framework for HAR. They used multiple CNN for feature learning and combined all three branches for getting a feature vector. With the combined three-layer data, they performed activity classification. Kim [24] used a sparse convolutional layers for resolving the tradeoff between accuracy and interoperability. The proposed framework removes the redundant and irrelevant signals, making the classifier learn optimally.

Dua et al. [25] proposed a multiinput CNN-GRU-based HAR system using wearable sensors. They employed CNN and gated recurrent unit (GRU) for automatic feature extraction and selection. With the help of their multiinput feature engineering architecture, they managed to achieve an average accuracy of 97.21% with the WISDM [15] dataset. Basly et al. [26] used CNN for automatic feature engineering and integrated it with a support vector machine (SVM) for activity classification. Al-qaness et al. [27] proposed an attention-based Multi-ResAtt for wearable sensor-based HAR. They incorporated a BIGRU for attention network for learning global features from the datasets. In [28], the authors integrated handcrafted features with a single-layer feed-forward network for assisting the LSTM network.

In [29], the authors used three-layer CNN for feature extraction and mapping. Using a fully connected LSTM model with batch normalization, they classified different human activities.

TABLE I
PUBLICLY AVAILABLE SENSOR-BASED HAR DATASET

Dataset Name	# of ADLs	# of Subjects	Sensors Used	Mounting Position	Data Collection Environment
HARSENSE [4]	6	12	A,G	Waist, Pocket	Controlled
WISDM [15]	18	51	A,G	Pockets, Hand	Controlled + Uncontrolled
UCI-HAR [10]	6	30	A,G	Dynamic	Controlled
MobiAct [16]	12	66	A,G	Front Pockets	Controlled
Unimib SHAR [17]	9	30	A	Front Pockets	Controlled
PAMAP2 [18]	12	9	A,G,T	Chest, Wrist, Ankle	Controlled
OPPORTUNITY [19]	6	4	A,G,M,AM	Hip, Leg, Shoes, Upper body	Controlled
mHEALTH [20]	12	10	A,G,M,ECG	Ankle, Wrist, Chest	Uncontrolled
MOTIONSENSE [21]	6	24	A,G	Front Pockets	Uncontrolled
OWN DATASET	**6**	**20**	**A,G**	**Front Pockets**	**Uncontrolled (In daily living Env.)**

*A = Accelerometer, G = Gyroscope, M= Magnetometer, AM = Ambient Sensor, SS = Stretch Sensor, T = Temperature, ECG = Electrocardio-gram
The bold entities to highlight our generated dataset attributes.

The authors in [30] used principal component analysis (PCA) for combining and weighted summation of spatial and temporal features. Later with the help of LSTM, they recognized various activities. In [31], the authors experimented with data augmentation and LSTM networks for HAR using Wi-Fi signals. They incorporated eight-channel state information (CSI) to mitigate all the inconsistent human activity and subject-dependent issues. The manufactured data they generated with CSI approach plays a vital role in improving overall model accuracy, and managed to get an average accuracy of 90%.

III. PROPOSED MODEL

A. Data Acquisition

Numerous sensor-based HAR datasets as referred in Table I are publicly available and have good quality data at different sampling frequencies and a sensors mix. Among them, very few datasets [20], [21] consist of the time-series ADLs data in an uncontrolled environment. Because of the lack of nonsimulated data, real-time activity recognition suffers and hampers overall system performance.

$$F_{nX16} = \begin{Bmatrix} a1_{x_0} & a2_{y_0} & a3_{z_0} & \dots & a14_{y_0} & a15_{y_0} & a16 \\ a1_{x_1} & a2_{y_1} & a3_{z_1} & \dots & a14_{y_1} & a15_{y_1} & a16 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ a1_{x_n} & a2_{y_n} & a3_{z_n} & \dots & a14_{n} & a15_{y_n} & a16 \end{Bmatrix} \quad (1)$$

$$Dataset_{nX17} = \{F \quad ClassLabels\}. \quad (2)$$

We have created a state-of-art HAR dataset in an uncontrolled environment to encounter this problem. ADLs raw data are collected with the help of four inbuilt smartphone sensors named *Samsung Galaxy F62, Samsung Galaxy A30 s, Poco X2, and One Plus 9 Pro*. As all these devices are of different parties and have different sensor manufacturers, a good amount of variable data will be collected and will increase the robustness. The sensor module was mounted in front pockets (vertically—phone earpiece side up) of different subjects. All the subjects were asked to perform various activities as freely as possible to get their natural ADLs data. The sensor mounting position with different sets of sensors incorporated is shown in Fig. 1(a).

A total of 20 subjects were considered of different ages, gender, weight, and heights. All the subjects were adults and

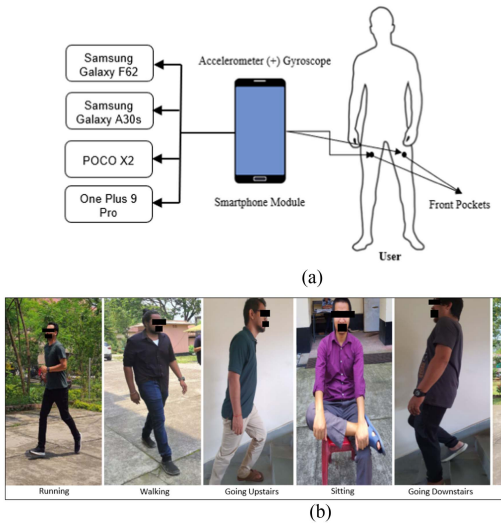

Fig. 1. Data acquisition module with incorporated sensor module and user performed activities. (a) Sensor mounting position with incorporated sensors for data collection. (b) Different ADLs performed by the users in uncontrolled environment.

were above 24 years old. The height and weight of different subjects are defined as $Weight(W_t) = 60 \pm 94\ kgs$ and $Height(H_t) = 165 \pm 186\ cms$. The dataset has six ADLs—*sitting, walking, standing, running, going upstairs and downstairs*. All activity was done as per the individual capacity and convenience. Activities like running, walking, sitting, and standing were done outside and upstairs and downstairs activities were done in a departmental building. A sample of the different users performing distinct activities is shown in Fig. 1(b). The sensor considered for collection of time-series data are accelerometer and gyroscope at 100 Hz and the formation of dataset with its generalized structure is shown in (2). The considered attributes in 3-D axis are *Acceleration due to Gravity* $(a1, a2, a3)$, *Linear Acceleration* $(a4, a5, a6)$, *Gravity* $(a7, a8, a9)$, *Rotational Rate* $(a10, a11, a12)$, *Rotational Vector* $(a13, a14, a15)$, *and* $\cos(a13)$ *as* $a16$ as described in (1).

B. Data Preprocessing

After data collection, we preprocessed the raw sensor data to make it robust, easy to use and manageable. First, we removed noise from the generated dataset. We removed each activity's top 100 and bottom 100 data instances as starting, stopping,

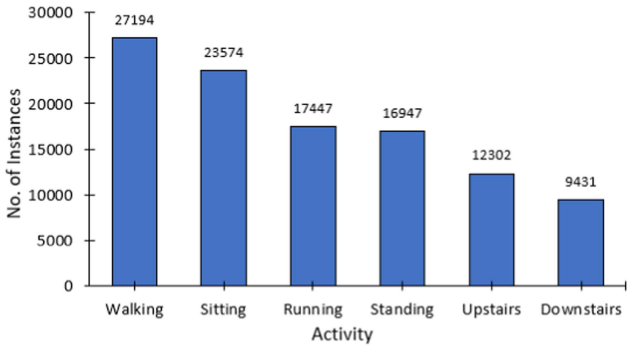

Fig. 2. Frequency count (class imbalance) of different activities in dataset.

keeping in, and taking out the smartphone module from the mounting position will take an adequate amount of time and is not related to any human activity. Then we checked for null and duplicate instances with the help of the *pandas* library. Our dataset does not have null instances but has a favorable number of identical values. We did not remove the duplicate instances as every activity class has an ample number of duplicate values and will not cause any bias to each other.

In the following phase of data preprocessing, we checked for class imbalancing and found out that our dataset has a notable class imbalance problem, as shown in Fig. 2. It is present as every subject was free to do its activity in their own way and they performed it based on their individual capacity and environment. Upstairs and downstairs activity instances are less in number as the data collection environment has a limited number of staircases. Also, as the users were allowed to do the activity freely, they all performed different activities for different time periods, leading to good class mismatch when all the user's data are combined in a single dataset. Handling or altering the duplicate values and imbalance problem will be incorrect as it will not mitigate the uncontrolled scenario. So we did not handle duplicate values and class imbalance problems to make our experiment challenging. Finally, we encoded the different activities using *scikit-learn LabelEncoder()* to transform them into numerical labels so that they can be fed to the classifier for model training.

C. Model Building and Selection of Evaluation Metrics

After completing data preprocessing, we created an adaptive batch-size-based CNN-LSTM model for classifying ADLs. In the first phase, we developed a time-interval-based sliding window selection scheme for getting the best input and output pair for model training. The same scheme is described in Algorithm 1. Time Period (T) and Stride (S) is calculated using (3) and (4), respectively. The overlapping window of the sliding window scheme is set to $\beta(10 \leq \beta \leq 100)$

$$Time\ Period\ (T) = \frac{1}{Frequency(F)} \tag{3}$$

$$Stride\ (S) = \frac{1000}{T}. \tag{4}$$

Once the dataset is segmented and reshaped based on the data acquisition time-period and strides, we created a hybrid

Algorithm 1: Time-Interval-Based Sliding Window Scheme.

Input 1: Dataset (D), Time period (T) in milli sec, Stride (S), # of features (n)
Output: Reshaped Data Segments.

procedure RESHAPEDATASEGMENTS D, T, S, n
  for $\{i = 0; i \leq (len(D) - T); S++\}$ **do**
    $f1 = feature_1.values[i : (i + T)]$
    $f2 = feature_2.values[i : (i + T)]$
    $f3 = feature_3.values[i : (i + T)]$
    .
    $fn = feature_n.values(i : (i + T)]$
    $Segment = Append([f_1, f_2, f_3, \ldots, f_n])$
  end for
  Return$(array(Segment.reshape(-1, T, n)))$
end procedure

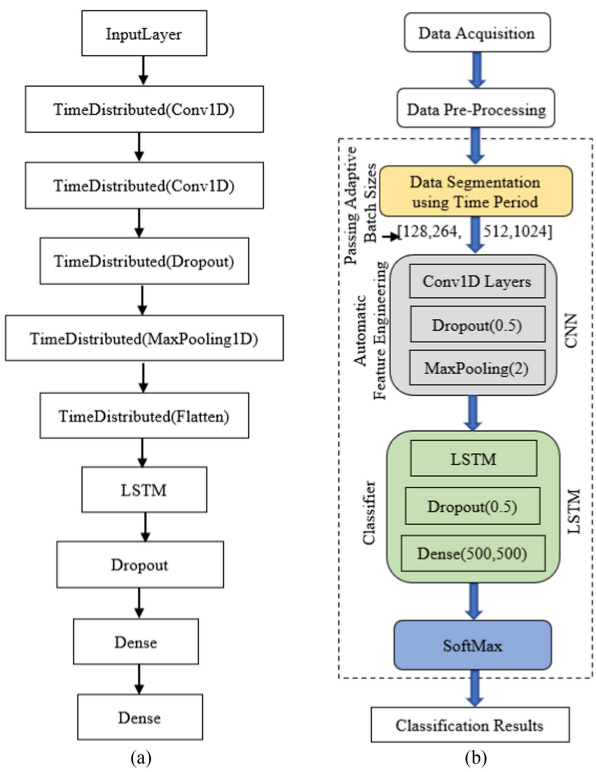

Fig. 3. Details of the proposed adaptive CNN-LSTM model. (a) Workflow the proposed model. (b) Proposed hybrid CNN-LSTM model.

CNN-LSTM model for automatic feature engineering and activity recognition as shown in Fig. 3. The hybrid CNN-LSTM model is chosen because of the efficient spatial and temporal feature handling by CNN and LSTM, respectively. CNN uses its convolutional operations for extracting low-level spatial features like max, min, average, etc., using one or many pooling layers and finding the most appropriate feature space for different human activities. LSTM uses its memory for storing the current and past stage information and extracts the temporal features for the new window to get the activity prediction. Along with this, we introduced an adaptive batch-size-based-model training method using iterative training and validation. It uses distinct batch

sizes for specific user-defined iterations assisting better model generalization on the validation set. Using big batch sizes indeed reduces computational time but converges the model to sharp minima, leading to poor generalization. On the contrary, small batch sizes generalize well but take too much computational time for model training as number of training iteration will be more. The proposed method handles the above two problems using both the small and large batch size model training, which will help in learning the weights better by picking up suitable samples from the various batch sizes in an iterative manner. The algorithm for the proposed model is described in Algorithm. 2.

Along with our proposed model, a few other classifiers like dense neural network (DNN), LSTM, and CNN-LSTM and two publicly available datasets—mHealth [20] and MotionSense [21] are also employed to compare and benchmark our proposed model in terms of accuracy, computational times, and other evaluation metrics. The incorporated evaluation metrics for benchmarking are calculated with the help of the confusion matrix and are defined as follows.

1) Precision (P): It is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p), and thus, calculated as: $P = \frac{T_p}{T_p+F_p}$
2) Recall (R): It is the actual positives of the classification model that is defined as the total number of true positives (T_p) over the total number of true positives (T_p) and false negatives (F_n), and thus, calculated as: $R = \frac{T_p}{T_p+F_n}$
3) F1-Score (F1) calculates the mean between the *Precision (P) and Recall (R)* and is one of the popular evaluation metrics, and thus calculated as: $F1 = 2 * \frac{P*R}{P+R}$
4) Accuracy (A) is the most frequently used performance metric in HAR system and is defined as the ratio of correctly predicted class (*total true instances*) of the total number observations (both positive and negative classes), and thus, it is calculated as: $A = \frac{T_p+T_n}{T_p+T_n+F_p+F_n}$

Note: The same formulas are used for multiclass classification using one versus rest classification scheme.

IV. Results and Discussion

All the model training, testing, and validation are done on single system hardware with 32 GB (31.6 GB usable) 2666 MHz RAM, Intel Xeon W-2133 3.60 GHz CPU, 1.8 TB HDD, and NVIDIA Quadro P2000 (5 GB) GPU. GPU is not utilized for model training and testing as we have enough computational power with our central processing unit. We have used python programming language (version - 3.9.7) for building our deep learning model.

A. Performance Comparison With Various Deep Learning Models

After all the data preparation, preprocessing and model generation, we trained and tested our model. Classifying different human activities of our dataset with the proposed model, we achieved the highest accuracy of 99.29% and an average accuracy of 98% with the hyperparameters described in Table II. While training the model, we first got volatile validation loss till 50 epochs (with 128 batch size), but after iterative training with adaptive batch sizes perishing, the validation, as well as training loss, stabilizes and goes to minimal. Also, the accuracy of training and validation data is remarkably high and stable after complete model training, as shown in Fig. 5(d)–(g). Comparing the proposed model with other deep learning models, our model achieves better accuracy with an average accuracy aperture of $2 \pm 5\%$. Our proposed model achieves the highest accuracy, and the accuracy cum F1-score comparison between different incorporated models is shown in Fig. 4.

To see the performance measurement of our model with benchmark models, we visualize the accuracy and loss graph.

Algorithm 2: Proposed CNN-LSTM Model.

Input 1: Reshaped Segmented Dataset (D_1), Class_Labels= C_L, Test_Size (T_s), Batch_Sizes (B_S) = [128, 264, 512, 1024]
Output: Trained Model.

procedure TRAIN TEST SPLITD_1, T_s, C_L
  $X\_train, X\_test, Y\_train, Y\_test = T_s$
  $Stratify = C_L$
end procedure
procedure CNN_LSTM($lstm\_neurons$, $dense\_neurons, drop\_out$)
  $model \leftarrow Sequential()$
  $model \leftarrow add(TimeDistributed(Conv1D),$
  $input\_shape = (None, n\_length, n\_features)))$
  $model \leftarrow add(TimeDistributed(Conv1D()))$
  $model \leftarrow$
  $add(TimeDistributed(Dropout(drop\_out)))$
  $model \leftarrow$
  $add(TimeDistributed(MaxPooling1D(2)))$
  $model \leftarrow add(TimeDistributed(Flatten()))$
  $model \leftarrow add(LSTM(lstm\_neurons))$
  $model \leftarrow add(Dropout(drop\_out))$
  $model \leftarrow$
  $add(Dense(dense\_neurons, activation = relu))$
  $model \leftarrow$
  $add(Dense(num\_classes, activation =$
  $softmax))$
  $model \leftarrow compile()$
  return $model$
end procedure
Reshaping data into time steps of sub-sequences
n_stride, n_length
Reshape $(X\_train\ and\ X\_test)$ as
$(X\_train\_cnn\ and\ X\_test\_cnn)$
# Initializing CNN_LSTM model
CNN_LSTM($lstm\_neurons, dense\_neurons, drop\_out$)
# Calling Adaptive CNN_LSTM
$cnn\_lstm = \{\}$ # Defining dictionary for saving iterative model training.
for $\{i\ in\ B_S\}$ **do**
  $B_S \leftarrow i$
  $epochs \leftarrow 50$
  $cnn\_lstm[] \leftarrow CNN\_LSTM.fit(B_S)$
end for

TABLE II
SUMMARY OF HYPERPARAMETERS USED IN OUR PROPOSED CNN-LSTM MODEL

Stage	Metric	Used Parameter
Architecture	Convolution	Conv1D
	Filter	64
	Kernal_size	3
	drop_out	0.5
	MaxPooling	MaxPooling1D(2)
	CNN_activation	relu
	LSTM_activation	relu
	output_activation	softmax
	LSTM_dense	500
	output_dense	500
Training	epoch	(50*4)
	batch_size	adaptive [128, 256, 512, 1024]
	loss_function	categorical_crossentropy
	optimizer	adam

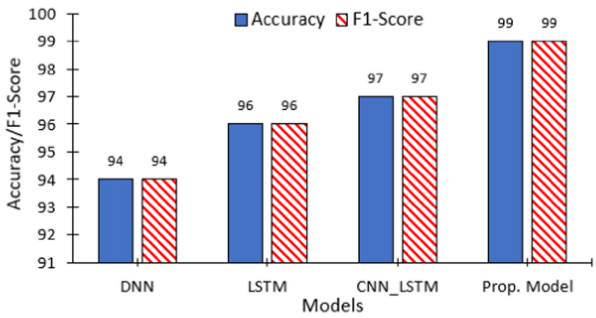

Fig. 4. Accuracy cum F1-score comparison of different models.

Comparing the graphs, we found out that the benchmark models have volatile training and validation loss compared to the proposed model. Also, incorporated models have sudden high validation loss at some point of iterations in model training. The model takes considerable training iterations to get stable and generalize the data again. Because of this, the performance degrades while testing the model. In Fig. 5(a) of DNN model accuracy and loss graph has steep loss on validation data and it increases with the training. LSTM has unstable training validation data loss and after considerable iterations of model training, the loss is increasing as shown in Fig. 5(b). This happens because of the lack of inference capabilities at different iterations. CNN-LSTM yields better accuracy than DNN and LSTM but could not able to cope up with the proposed model as its validation loss is not stable as shown in Fig. 5(c). The same hyperparameter of our proposed model is kept for benchmark CNN-LSTM and DNN and LSTM hyperparameter were tuned for the best performance. DNN, LSTM, and normal CNN-LSTM are chosen for model comparison as they all have different learning architecture and works differently. Comparing our proposed model with them, gives an overall insight into different model learning paradigms and suggests what to choose for a prominent HAR system.

B. Analysis of Class Imbalance and Computational Time

In Section III-B, the class imbalancing problem was not taken care of and we stated that our model can efficiently handle the imbalanced data. Observing the Precision (P) and Recall (R) values, it can be seen that the proposed model yields better results compared to the benchmark models and in less computational time as well. The effect of different batch size from small to big passes the adequate number of data instances of each class to the model and handles the imbalancing problem and its effect can be seen in confusion matrix as shown in Fig. 6. One more reason for computational efficiency is the use of ReLu and Softmax activation functions for enumerating the feature vector and output, respectively. ReLu uses its partial neuron activation policy by making the network sparse using (5) and decreases the overall computational overhead of the network. Softmax is always considered very efficient in output layers for multiclass classification and as it converts its input into a vector of probability distributions that determines the potential outcomes. Using the mathematical function in (6), it calculates its nonlinear output and optimizes CPU utilization using its straightforward output calculation

$$f(x) = \max(0, x_i) \begin{cases} x_i & if \;\; x_i \geq 0 \\ 0 & if \;\; x_i < 0 \end{cases} \tag{5}$$

$$f(x_i) = \frac{\exp(x_i)}{\sum_1^j \exp(x_j)}. \tag{6}$$

Table III shows the detailed comparison of different evaluation metrics with computational times. The macro average (M_{Avg}) and weighted average (Wt_{Avg}) is calculated for classified P, R, and F1 results and is defined in (7), (8), where w is support count, x is the classified P, R, F1 values, and N is the number of classes. Computational time (C_t) is calculated with the help of *process_time()* python function which only calculates the CPU utilization time and it defined as the sum of model training, validation, and testing time as shown in (9), where T_r is training time, T_v is the validation time and T_s is the testing time. The loss in testing data is very high for DNN and LSTM models as there is a need for good feature analysis handled by CNN in hybrid models like CNN-LSTM. Comparing it to the hybrid models like CNN-LSTM and our proposed CNN-LSTM, the loss is small and can be further optimized because of its capability to handle both spatial and temporal features.

$$M_{Avg} = \left[\frac{\sum_{i=1}^{n} x_i}{N}\right] \tag{7}$$

$$Wt_{Avg} = \left[\frac{\sum_{i=1}^{n} x_i \, w_i}{\sum_{i=1}^{n} w_i}\right] \tag{8}$$

$$C_t = [T_r + T_v + T_s]. \tag{9}$$

C. Performance Comparison of Standard Datasets With Benchmarks

Two publicly available datasets named mHelath [20] and MotionSense [21] are incorporated for validating our proposed model. Table IV shows the detailed performance metrics achieved by our model on the incorporated datasets. All the standard datasets achieved an average accuracy of above 99% with our proposed model and outperformed the previous benchmark works as compared in Table V. Cross-verification of the performance results with fivefold cross-validation is also done, and the

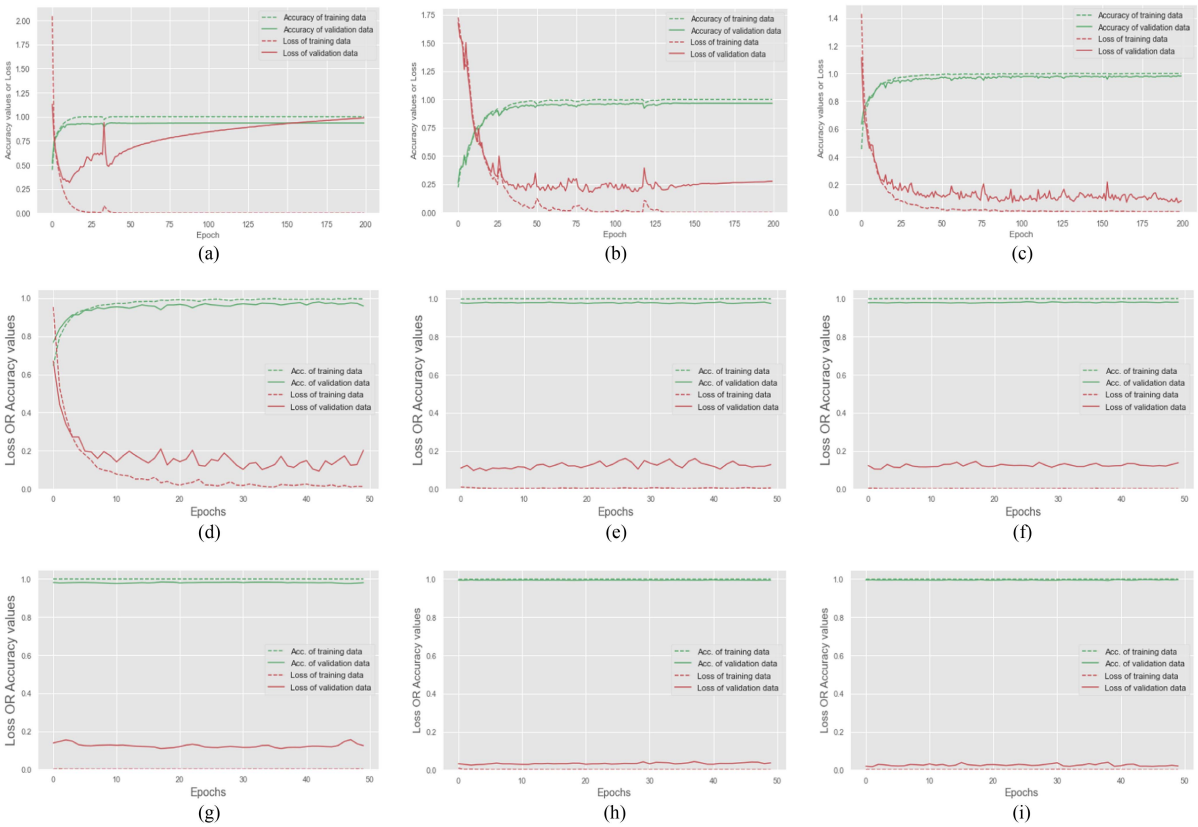

Fig. 5. Model accuracy and loss of proposed CNN-LSTM model on incorporated datasets and other classifiers. (a) DNN with batch_size 1024. (b) LSTM with batch_size 1024. (c) CNN-LSTM with batch_size 1024. (d) Prop. model first 50 epochs with batch_size 128. (e) Prop. model second 50 epochs with batch_size 256. (f) Prop. model third 50 epochs with batch_size 512. (g) Prop. model final 50 epochs with batch_size 1024. (h) Prop. model final 50 epochs with batch_size 1024 for mHealth dataset. (i) Prop. model final 50 epochs with batch_size 1024 for MotionSense dataset.

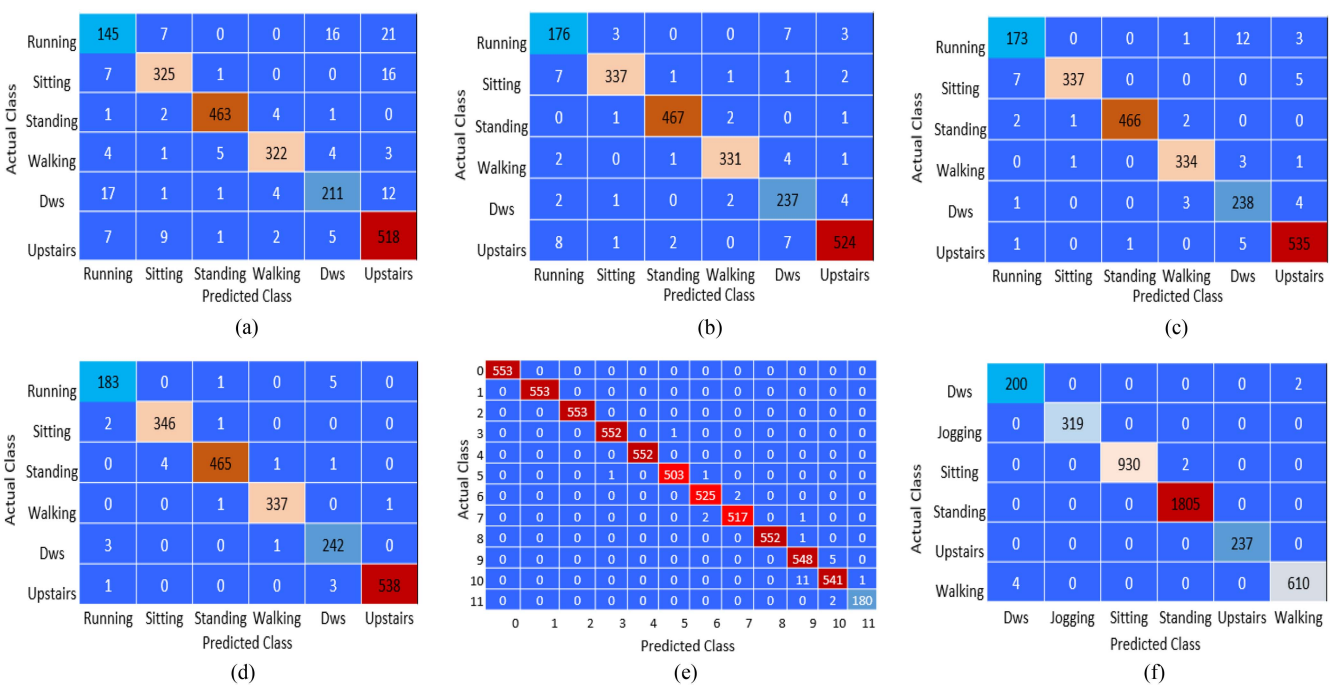

Fig. 6. Confusion matrix of different benchmark models with proposed CNN-LSTM model. (a) Confusion matrix for DNN. (b) Confusion matrix for LSTM. (c) Confusion matrix for CNN-LSTM. (d) Confusion matrix for proposed CNN-LSTM. (e) Confusion matrix for proposed CNN-LSTM on mHealth dataset. (f) Confusion matrix for proposed CNN-LSTM on MotionSense dataset.

TABLE III
COMPARISON OF DIFFERENT EVALUATION METRICS OF OUR PROPOSED MODEL WITH BENCHMARK MODEL

Models / Metric	Acc (%)	Avg. Method	Precision	Recall	F1-Score	Loss (Testing Data)	Comp. Time (in Frac. Sec.)	Acc. (K fold k=5)
DNN	94	Macro Avg.	0.93	0.93	0.93	0.890 ± 0.893	16015 ± 16105	93.89
		Weighted Avg.	0.94	0.94	0.94			
LSTM	96	Macro Avg.	0.95	0.95	0.95	0.224 ± 0.259	49316 ± 51402	95.73
		Weighted Avg.	0.96	0.96	0.96			
CNN-LSTM	97	Macro Avg.	0.96	0.96	0.96	0.184 ± 0.212	8623 ± 9812	96.96
		Weighted Avg.	0.97	0.97	0.97			
Prop. Model	99	Macro Avg.	0.98	0.98	0.98	0.080 ± 0.136	8511 ± 8715	99.01
		Weighted Avg.	0.99	0.99	0.99			

TABLE IV
PERFORMANCE OF PROPOSED MODEL ON STANDARD DATASETS IN UNCONTROLLED ENVIRONMENT

Datasets	Acc. (%)	Avg. Method	P	R	F1
Own Dataset	99.29	Macro Avg.	0.98	0.98	0.98
		Weighted Avg.	0.99	0.99	0.99
mHealth [20]	99.51	Macro Avg.	1	0.99	1
		Weighted Avg.	1	1	1
MotionSense [21]	99.85	Macro Avg.	1	1	1
		Weighted Avg.	1	1	1

TABLE V
ACCURACY COMPARISON OF PROPOSED MODEL ON STANDARD DATASETS IN UNCONTROLLED ENVIRONMENT

Dataset	Benchmark Work	Accuracy (%)
mHealth [20]	Semwal et al. [32]	94
	Gracia et al. [33]	94.8
	Javeed et al. [34]	93.3
	Burns et al. [35]	96.7
	Singh et al. [12]	97.65
	Proposed Model	**99.6**
MotionSense [21]	Bautet et al. [36]	92
	Sharshar et al. [37]	95.05
	Choudhury et al. [4]	96
	Proposed Model	**99.8**
Own Dataset	**Proposed Model**	**99.21**

The bold values represents the proposed model performance.

model managed to procure accuracies above 99% throughout the experiment. Our model is performing better on public datasets as the huge number of data instances is available for model training. Among the two incorporated datasets, MotionSense is highly imbalanced. With this problem as well, the model achieved 100% precision and recall, which shows the robustness of the adaptive batch size and CNN-LSTM model. The loss of the model on both of the public datasets is also very low as it is not losing any training and validation accuracy, as shown in Fig. 5(i) and (j). The confusion matrix for both the public dataset with our proposed model is shown in Fig. 6(e) and (f).

V. CONCLUSION

In this article, we experimented with the design, development, and implementation of a novel adaptive batch-size-based CNN-LSTM model for human activity recognition in an uncontrolled environment. A state-of-art HAR dataset was created in an uncontrolled environment to get the actual unsimulated data of different ADLs. The accuracy of our proposed model outperformed the incorporated benchmark models by a good performance as well as computational time margin and managed to achieve an average and highest accuracy of 98% and 99.29%, respectively. The proposed model also handled the class imbalance problem with ease as the precision and recall scores were high and consistent throughout the model training and testing phase. The inclusion of adaptive batch sizes while training the model reduced the computational time by 11.8422% (max) and reduced the training data loss. A time-interval based-sliding window scheme was also described for reshaping the data instances optimally using the time period of data collection. Moreover, we extrapolated that the hybrid deep learning algorithms that used CNN as the base feature extractor yielded better results and were computationally efficient. Finally, two publicly available datasets were exploited with the proposed model and managed to achieve an accuracy of 99.5% (mHealth) and 99.8% (MotionSense) and outperformed all the previous benchmarks.

In future, we will try to include more complex human activities with different sampling rate and mounting locations. Also, we will try to make a transfer learning-based CNN-LSTM approach for fine-tuning the overall model architecture.

ACKNOWLEDGMENT

Code and data availability: The code and dataset will be made available for academic and future research work on reasonable request. It will also contain support information, including sponsor, and financial support acknowledgment.

REFERENCES

[1] T.-H. Tan, J.-Y. Wu, S.-H. Liu, and M. Gochoo, "Human activity recognition using an ensemble learning algorithm with smartphone sensor data," *Electronics*, vol. 11, no. 3, 2022, Art. no. 322.
[2] S. Zhang et al., "Deep learning in human activity recognition with wearable sensors: A review on advances," *Sensors*, vol. 22, no. 4, 2022, Art. no. 1476.
[3] S. Mekruksavanich and A. Jitpattanakul, "LSTM networks using smartphone data for sensor-based human activity recognition in smart homes," *Sensors*, vol. 21, no. 5, 2021, Art. no. 1636.
[4] N. A. Choudhury, S. Moulik, and D. S. Roy, "Physique-based human activity recognition using ensemble learning and smartphone sensors," *IEEE Sensors J.*, vol. 21, no. 15, pp. 16852–16860, Aug. 2021.
[5] C. Shen, D. Nguyen, Z. Zhou, S. B. Jiang, B. Dong, and X. Jia, "An introduction to deep learning in medical physics: Advantages, potential, and challenges," *Phys. Med. Biol.*, vol. 65, no. 5, 2020, Art. no. 05TR01.

[6] V. Gavrishchaka, Z. Yang, R. Miao, and O. Senyukova, "Advantages of hybrid deep learning frameworks in applications with limited data," *Int. J. Mach. Learn. Comput.*, vol. 8, pp. 549–558, 2018.
[7] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "Trends in human activity recognition using smartphones," *J. Reliable Intell. Environ.*, vol. 7, no. 3, pp. 189–213, 2021.
[8] N. S. Khan and M. S. Ghani, "A survey of deep learning based models for human activity recognition," *Wireless Pers. Commun.*, vol. 120, no. 2, pp. 1593–1635, 2021.
[9] R. Mutegeki and D. S. Han, "A CNN-LSTM approach to human activity recognition," in *Proc. IEEE Int. Conf. Artif. Intell. Inf. Commun.*, 2020, pp. 362–366.
[10] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. IEEE Int. 21st Eur. Symp. Art. Neural Netw.*, 2013.
[11] S. Mohsen, A. Elkaseer, and S. G. Scholz, "Industry 4.0-oriented deep learning models for human activity recognition," *IEEE Access*, vol. 9, pp. 150508–150521, 2021.
[12] S. P. Singh, M. K. Sharma, A. Lay-Ekuakille, D. Gangwar, and S. Gupta, "Deep convLSTM with self-attention for human activity decoding using wearable sensors," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8575–8582, Mar. 2021.
[13] N. Tufek, M. Yalcin, M. Altintas, F. Kalaoglu, Y. Li, and S. K. Bahadir, "Human action recognition using deep learning methods on limited sensory data," *IEEE Sensors J.*, vol. 20, no. 6, pp. 3101–3112, Mar. 2020.
[14] S. Mekruksavanich and A. Jitpattanakul, "Smartwatch-based human activity recognition using hybrid LSTM network," in *Proc. IEEE Sensors*, 2020, pp. 1–4.
[15] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, 2011.
[16] C. Chatzaki, M. Pediaditis, G. Vavoulas, and M. Tsiknakis, "Human daily activity and fall recognition using a smartphone's acceleration sensor," in *Information and Communication Technologies for Ageing Well and e-Health*. Berlin, Germany: Springer, 2017, pp. 100–118.
[17] D. Micucci, M. Mobilio, and P. Napoletano, "UniMIB shar: A dataset for human activity recognition using acceleration data from smartphones," *Appl. Sci.*, vol. 7, no. 10, 2017, Art. no. 1101.
[18] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31314–31338, 2015.
[19] R. Chavarriaga et al., "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, 2013.
[20] O. Banos et al., "mhealthdroid: A novel framework for agile development of mobile health applications," in *Ambient Assisted Living and Daily Activities*. L. Pecchia, L. L. Chen, C. Nugent, and J. Bravo, Eds. Berlin, Germany:Springer, 2014, pp. 91–98.
[21] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," in *Proc. Int. Conf. Internet Things Des. Implementation*, 2019, pp. 49–58.
[22] 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 Appl.*, vol. 79, no. 41, pp. 31663–31690, 2020.
[23] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data," *Vis. Comput.*, vol. 38, no. 12, pp. 4095–4109, 2022.
[24] E. Kim, "Interpretable and accurate convolutional neural networks for human activity recognition," *IEEE Trans. Ind. Inf.*, vol. 16, no. 11, pp. 7190–7198, Nov. 2020.
[25] N. Dua, S. N. Singh, and V. B. Semwal, "Multi-input CNN-GRU based human activity recognition using wearable sensors," *Comput.*, vol. 103, no. 7, pp. 1461–1478, 2021.
[26] H. Basly, W. Ouarda, F.E. Sayadi, B. Ouni, and A. M. Alimi, "CNN-SVM learning approach based human activity recognition," in *Image and Signal Processing*, A. El Moataz, D. Mammass, A. Mansouri, and F. Nouboud, Eds. Berlin, Germany: Springer, 2020, pp. 271–281.
[27] M. A. A. Al-qaness, A. Dahou, M. A. Elaziz, and A. M. Helmi, "Multi-resAtt: Multilevel residual network with attention for human activity recognition using wearable sensors," *IEEE Trans. Ind. Inform.*, vol. 19, no. 1, pp. 144–152, Jan. 2023.
[28] Z. Chen, L. Zhang, Z. Cao, and J. Guo, "Distilling the knowledge from handcrafted features for human activity recognition," *IEEE Trans. Ind. Inf.*, vol. 14, no. 10, pp. 4334–4342, Oct. 2018.
[29] H. Wang et al., "Wearable sensor-based human activity recognition using hybrid deep learning techniques," *Secur. Commun. Netw.*, vol. 2020, 2020, Art. no. 2132138.
[30] Z. Zhang, Z. Lv, C. Gan, and Q. Zhu, "Human action recognition using convolutional LSTM and fully-connected LSTM with different attentions," *Neurocomputing*, vol. 410, pp. 304–316, 2020.
[31] J. Zhang et al., "Data augmentation and dense-LSTM for human activity recognition using Wi-fi signal," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4628–4641, Mar. 2021.
[32] 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.*, vol. 77, no. 11, pp. 12256–12279, 2021.
[33] K. D. Garcia et al., "An ensemble of autonomous auto-encoders for human activity recognition," *Neurocomputing*, vol. 439, pp. 271–280, 2021.
[34] M. Javeed, M. Gochoo, A. Jalal, and K. Kim, "HF-SPHR: Hybrid features for sustainable physical healthcare pattern recognition using deep belief networks," *Sustainability*, vol. 13, no. 4, pp. 1–28, 2021.
[35] D. Burns, P. Boyer, C. Arrowsmith, and C. Whyne, "Personalized activity recognition with deep triplet embeddings," *Sensors*, vol. 22, no. 14, 2022, Art. no. 5222.
[36] A. Boutet, C. Frindel, S. Gambs, T. Jourdan, and R. C. Ngueveu, *DySan: Dynamically Sanitizing Motion Sensor Data Against Sensitive Inferences Through Adversarial Networks*. New York, NY, USA: Association Computing Machinery, 2021, pp. 672–686.
[37] A. Sharshar, A. Fayez, Y. Ashraf, and W. Gomaa, "Activity with gender recognition using accelerometer and gyroscope," in *Proc. IEEE 15th Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2021, pp. 1–7.

Nurul Amin Choudhury received the B.Tech. degree in computer science and engineering from the Jawaharlal Nehru Technological University Hyderabad, India and the M.Tech. degree in computer science and engineering from the National Institute of Technology (NIT) Meghalaya, Shillong, India, in 2018 and 2021, respectively. Currently, he is working toward the Ph.D. degree in computer science and engineering with the National Institute of Technology (NIT) Silchar, Silchar, India.

His current research interests include human activity recognition, gait recognition, machine learning and deep learning applications, AI-IoT in health care, and smart learning algorithms.

Badal Soni received the B.Tech. degree in computer science and engineering from Rajiv Gandhi Technical University (formerly RGPV) Bhopal, Bhopal, India, in 2010, and the M.Tech. degree in computer science and engineering from the Indian Institute of Information Technology, Design and Manufacturing (IIITDM), Jabalpur, India, in 2012, and the Ph.D. degree in computer science and engineering under MoU with NIT Silchar and Indian Institute of Technology (IIT) Guwahati, Guwahati, India, in 2018.

He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, National Institute of Technology (NIT) Silchar, Silchar, India. He has teaching experience of over ten years in the area of computer science and information technology with special interest in image processing, machine learning, and language processing. He has authored or coauthored 60 publications in refereed journals, contributed books, and international conference proceedings. His research interests include image processing, medical image processing, machine learning application, fake news detection, and human activity recognition.

Dr. Soni is a Professional Member of various bodies like IEEE, ACM, IAENG, and IACSIT.