Multimedia Tools and Applications (2021) 80:35585–35604
https://doi.org/10.1007/s11042-020-10478-4

1166: ADVANCES OF MACHINE LEARNING IN DATA ANALYTICS AND VISUAL INFORMATION PROCESSING

Hybrid deep learning approaches for smartphone sensor-based human activity recognition

Vasundhara Ghate[1] · Sweetlin Hemalatha C[1]

Received: 17 April 2020 / Revised: 8 November 2020 / Accepted: 29 December 2020 /
Published online: 6 February 2021

Abstract

Human Activity Recognition (HAR) has become one of the most important research fields to achieve real-time monitoring of human activities for timely decision making in various applications like fall detection, elderly care etc. Now-a-days, most people use smartphones which come with various embedded inertial sensors like accelerometer and gyroscope to monitor acceleration and angular velocity. These smartphone-based sensors have proven to be cost-effective solution in identification of activities belonging to ADL (Activities of Daily Living). Various Machine Learning, Deep learning and hybrid models have been proposed and implemented for HAR. This paper also proposes various hybrid deep learning approaches which combine Deep Neural Networks with other models like LSTM (Long Short Term Memory) Model and GRU (Gated Recurrent Unit) for effective classification of engineered features from CNN (Convolutional Neural Network) Model. A novel architecture that integrates CNN with Random Forest Classifier (*DeepCNN-RF*) is proposed to add randomness to the model. The proposed models have been tested on publicly available HAR Datasets like UCI HAR and WISDM Activity Recognition Datasets. Experimental results show that the hybrid models outperform the state-of-the-art data mining, machine learning techniques in UCI HAR and WISDM with an overall maximum accuracy of 97.77% and 98.2% respectively.

Keywords HAR · ADL · Inertial sensors · LSTM · GRU · CNN · DeepCNN-RF

✉ Sweetlin Hemalatha C
sweetlinh@gmail.com

Vasundhara Ghate
vasundharavijay.ghate2016@vitstudent.ac.in

[1] School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, India

1 Introduction

Human Activity Recognition field has garnered huge importance as it allows monitoring and recognition of daily human activities in various applications like healthcare for elderly people, Smart Home systems, human fall detection and surveillance systems. Various methodologies have been proposed and implemented for HAR using Data Mining, Machine Learning and Deep learning techniques. The generalized architecture of an HAR system is depicted in Fig. 1.

The overall HAR System can be represented in two ways based on the type of input data fed to the system as shown in Fig. 2.

1.1 Computer vision-based

The input device in this system is generally a video camera which continuously captures the activities through videos. The input fed is in a format of images or video sequences. *Bux* et al. [4] have discussed the basics of HAR and compares the effectiveness of HAR systems between static images and video inputs. A thorough study of various manual feature engineering methods using Machine Learning and automatic feature extraction using Deep Learning for Computer Vision – based HAR approaches has been carried out. Though the video based HAR systems proven to be beneficial in various real-time applications like public security, surveillance, monitoring elderly patents etc., video data acquisition for continuous and real time monitoring, high deployment and implementation cost, privacy issues and constrained environmental factors pose a great challenge. *Ke* et al. [12] have presented an extensive survey of different HAR systems ranging from activity recognition of single person, multiple person, abnormal activity recognition etc. The authors have also covered various HAR application domains like healthcare, surveillance systems, entertainment domains etc. However, the aforementioned methods suffer various issues such as selection of viewpoint by multiple cameras, need of sophisticated algorithms to handle the occlusion problem.

1.2 Sensor-based

Sensor-based systems are found to be more effective than vision based system as they are not limited by the effect of environmental factors, device deployment cost etc. The following subsections discuss the two major sub-divisions of sensor-based HAR systems viz. wearable sensor based and smartphone based systems.

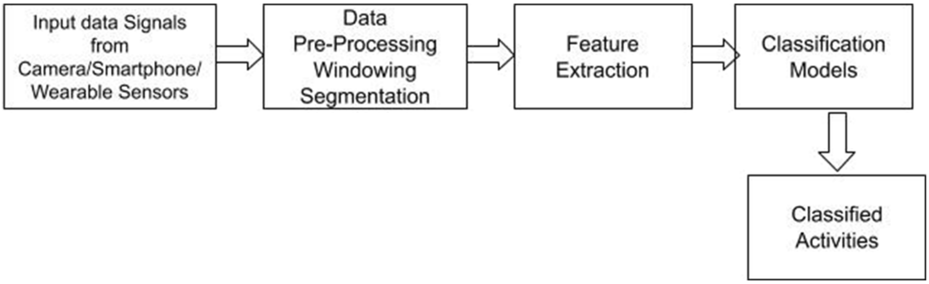
Input data Signals from Camera/Smartphone/ Wearable Sensors
Data Pre-Processing Windowing Segmentation
Feature Extraction
Classification Models
Classified Activities

Fig. 1 Generalized architecture of HAR systems

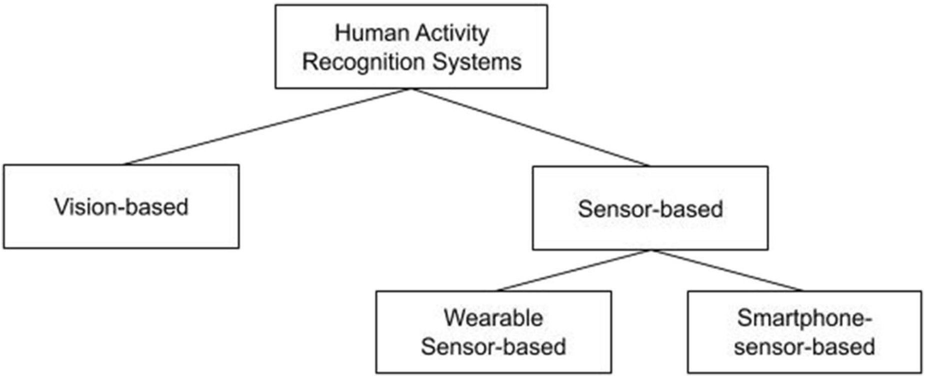

Fig. 2 Types of human activity recognition systems

1.2.1 Wearable sensor-based

Jordao et al. [11] have provided a comprehensive study of hand-crafted feature approaches to Convolution Neural Networks (CNN) for recognizing human activities based on wearable sensors worn on various parts of the body. *Mekruksavanich* et al. [17] have used smartwatch based sensors like accelerometer and gyroscope for sensing sitting activity of office workers and ensemble learning models for detecting the office workers syndrome problem. However, wearable sensor based systems are expensive as they demand the use of custom hardware.

1.2.2 Smartphone-based

Now-a-days, smartphones come with embedded sensors like accelerometer and gyroscope which do not require any additional hardware or installation. Lots of research works have reported Smartphone-based sensor data input for implementing HAR. *Davide* et al. [2] and *Mandong* et al. [16] have used hand crafted features and experimented with various Machine learning algorithms like Nearest Neighbors [24], Random Forest and SVM(Support Vector Machine) [23] for recognizing human activities. Kwapisz et al. [14] have discussed the advantages of using smartphone sensors for human activity recognition

A review on various machine learning algorithms has been presented in [3, 27] which includes Random Forest, SVM, Naive Bayes etc., and results showed that the Random Forest classifier has performed better compared to other approaches. Principal Component Analysis (PCA) [10], [26] has been used for reducing the dimensionality of features which are then fed to the multi-layer perceptron to classify the human activities. Nevertheless, all machine learning techniques require the need of hand-crafted features for human activity recognition, which poses limitations on the accuracy of the HAR system. On the other hand, deep learning approaches overcome the limitations of hand-crafted features by capturing the complex non-linear interactions between the features without intervention.

The rest of the paper is organized as follows. Section II discusses the related work in the field of Deep Learning methods for effective HAR. Section III presents the proposed methodologies including the model architecture and implementation steps. Section IV presents the description of the data sets used and the results obtained under the Experimental setup and Result Analysis. Section V provides comparative analysis of various existing deep learning approaches and the proposed hybrid approaches based on implementation results. Section VI draws conclusion and future directions of the proposed study.

2 Related work

Several research works have discussed the need and effectiveness of applying various deep learning approaches like CNN (Convolutional Neural Networks) for recognizing human activities. *Zeng* et al. [29] *Ronao* et al. [22] have presented the use of ConvNet, a deep Convolutional Neural Network based approach which extracts robust and relevant features from input time series data automatically with each added layer in the network. It also proved beneficial in achieving efficient classification of moving activities.

Sojeong et al. [6] have proposed a multi-modal 2D kernel CNN-based model on benchmark datasets available for HAR. The approach showed improved performance compared to various 1D CNN approaches and data mining techniques on existing HAR datasets.

Qingzhong et al. [15] have presented smartphone-based sensors based HAR which not only include the data related to human movement but also phone movement. It focused on binary classification based on created events. Applying same approach for datasets having multiple labels for multi-class classification is a complex task and yet to be tested.

The use of Recurrent Neural Network (RNN) based approaches which involve use of LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) have been experimented in many works as they are most suitable for time-series based sensor signals as input data. *Okai* et al. [19] have discussed HAR using RNN with data augmentation which was compared with other deep learning algorithms like LSTM and GRU. The advantages of individual deep learning models are proven to be more useful if combined together as hybrid deep learning approaches as reported in few works [1].

A hybrid DeepConvLSTM Model [20] has been proposed on well-known existing HAR datasets and the technique outperformed the performance of individual models on same datasets as it combined the feature of temporal data analysis of LSTM and automatic feature engineering feature of CNN. *Wang* et al. [28] presented a survey on various deep learning approaches applied for effective HAR which includes basic individual and hybrid models like ConvLSTM as discussed earlier. The survey took into consideration four existing and commonly used HAR datasets: Opportunity I, UCI Smartphone, Opportunity II and Skoda datasets. From the comparative analysis, it can be seen that hybrid approaches provided higher accuracy value as compared to standalone implementation of deep learning models.

Leveraging the advantages of hybrid deep learning models, this paper also discusses a few hybrid deep learning approaches which uses CNN for efficient feature extraction before being fed to the GRU/LSTM networks for further classification tasks. A novel hybrid approach: CNN-RF is proposed which takes into account the effectiveness of an ensemble learning methodology with randomness property which can be seen in Random Forest Classifier for final classification based on the features extracted using CNN model.

3 Proposed work

This work aims at applying Deep Learning techniques and proposing various hybrid deep-learning approaches for HAR like discussed in [3] that combine the automatic feature extraction characteristic of CNN (Convolutional Neural Networks) and classification using other models like Long Short Term Memory, GRU and Random Forest. Two datasets are considered for analysis: Smartphone-based Human Activity Recognition by UCI Repository [12] and WISDM's Smartphone Sensor-based datasets [13].

Data Acquisition from Smartphone-based embedded sensors involves collection of raw signals from sensors like accelerometers for acceleration, gyroscopes for angular velocity. The accelerometer and gyroscope generates raw x, y and z axis acceleration and angular velocity signals collected on a time-series basis.

3.1 Data pre-processing

The raw data collected from these sensors also contains noise along with important information. Thus, there is a need to filter the data by minimizing the noise. Various filters such as median filters and Butterworth filters are used for filtering noise from raw signals. After filtering, the input sensor data stream is divided into different individual segments by using a commonly used windowing technique like sliding window. Generally, in segmentation process, a fixed/variable length window or a fixed/varied count of sensor events is shifted through the input stream by overlapping/non-overlapping allowed between the adjacent segments as discussed in [13]. In UCI dataset, 50% overlapping of windows was applied for segmenting the signals.

The graphical representation of the input dataset signal values based on activity type and user-id/subject for UCI HAR dataset can be seen in Fig. 3a and b respectively. The graphs for input raw signals along x, y, z axis of accelerometer for each activity in UCI dataset is shown in Fig. 4a to f. The raw signals need to go through filtering process to remove unwanted noise and get only important information. After filtering, the signals are segmented based on the windowing technique called sliding window protocol as discussed in [21].

The next step is feature extraction where important features are extracted from the raw sensor data segments and represented as feature vectors. Then, a classification model is trained based on these feature vectors. The last step is to use the built classifier to predict the activity of a stream of sensor data. This study focusses on the use of the built-in tri-axial accelerometer and gyroscope. Therefore, there are two types of data generated by sensors in a smartphone.

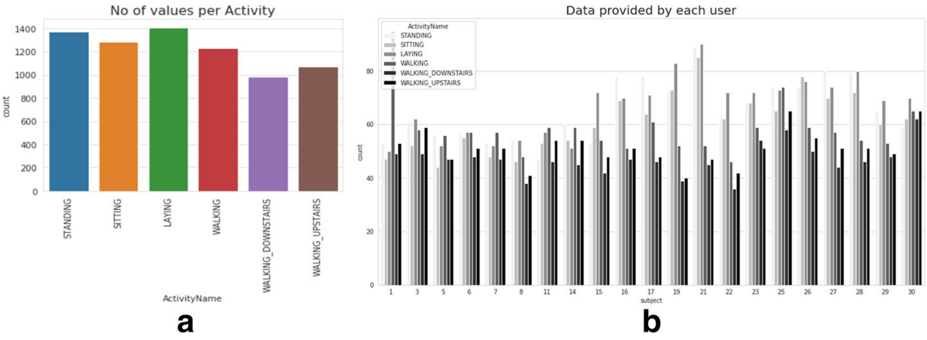

Fig. 3 **a** Training examples based on activity type **b** Training examples based on User ID (UCI HAR dataset).

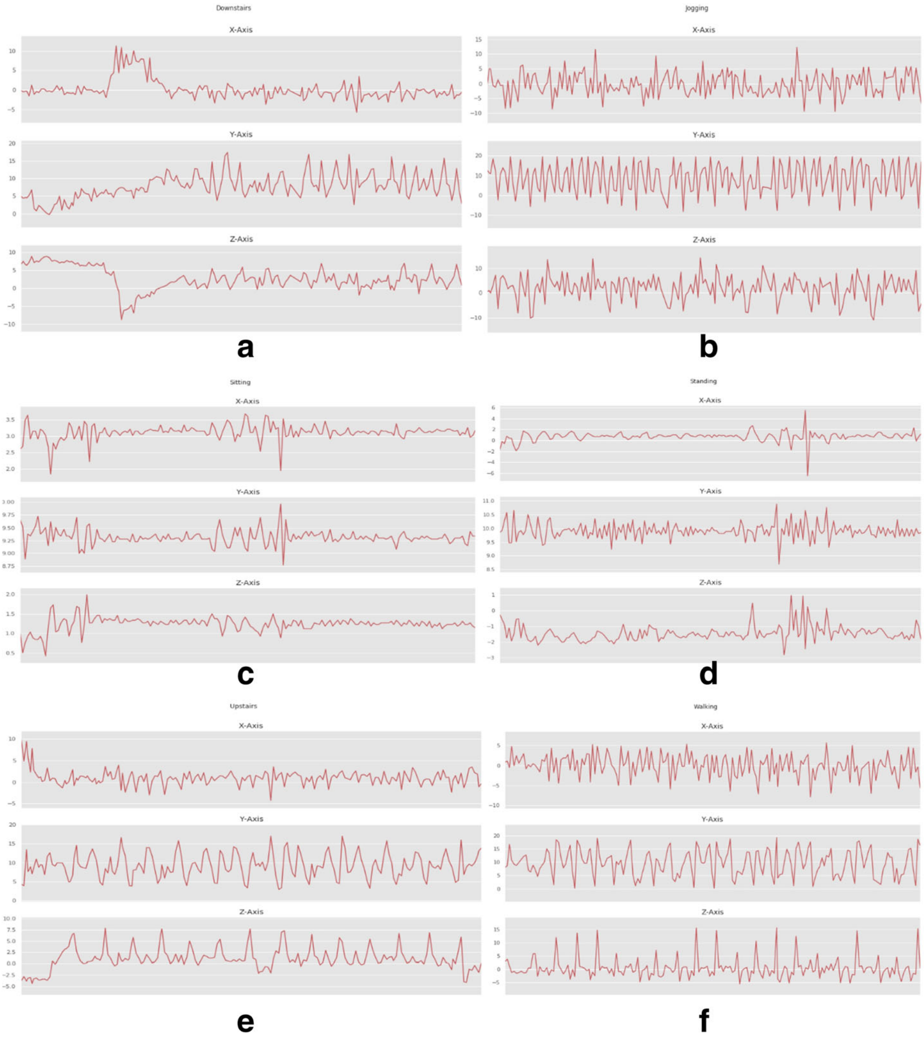

Fig. 4 Acceleration data along x, y, z axis for all activity types (UCI HAR Dataset) (**a**) Walking Downstairs (**b**) Jogging (**c**) Sitting (**d**) Standing (**e**) Walking upstairs (**f**) Walking

3.2 Feature extraction and selection

After segmentation, it is essential to extract most important features based on frequency and time-domain like mean (μ), standard deviation (δ), entropy(S), correlation coefficient(ρ) etc. from the segments. These extracted features together will help in creating the input feature vector. From the created feature vector there is a possibility of high dimensionality and redundancy in features or possibility of presence of few irrelevant features. Therefore, applying feature selection is essential to select only important features from the input feature space which will minimize the overall training time while not affecting the overall performance. In the proposed work, Convolutional Neural Network is used for automated Feature Engineering.

3.3 Model creation and training

After feature engineering, the final set of reduced features are to be fed to the proposed models based on various deep learning models like Deep Neural Networks, Recurrent Neural Network, Convolutional Neural Networks. All these approaches make use of various Activation functions to predict the output of neural networks. The following subsections presents the various types of non-linear activation functions used widely in Deep Learning approaches:

3.3.1 Sigmoid/logistic function

Sigmoid Function is a traditional activation function which is mostly used for binary classification problems. This function suffers from various limitations like vanishing gradient issue, not being zero-centered and also expensive with respect to computation. Figure 5 depicts the sigmoid function and Eq. 1 represents the corresponding expression.

$$f(x) = \frac{1}{1 + e{-}x} \tag{1}$$

3.3.2 Tanh function

It is similar to sigmoid function but it is more symmetric around the origin. Thus, the range of values represented by this function is between 1 and − 1. Figure 6 depicts the sigmoid function and Eq. 2 represents the corresponding expression.

$$\tanh(x) = 2sigmoid(2x){-}1 \tag{2}$$

3.3.3 ReLU (rectified linear unit)

It is the most popular and widely used non-linear activation function as it does not activate all the neurons at the same time. The neuron is deactivated only if its value of linear

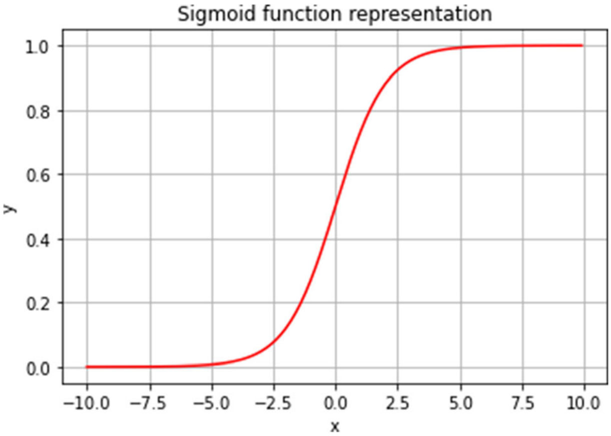

Fig. 5 Graphical representation of Sigmoid function

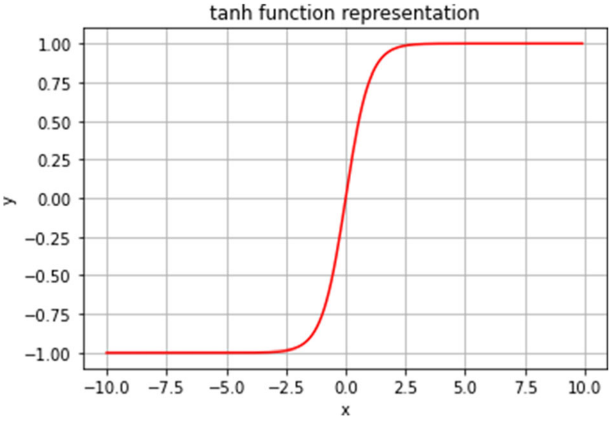

Fig. 6 Graphical representation of tanh function

transformation is less than 0. Figure 7 shows the graphical representation of ReLU (Rectified Linear Unit) function and Eq. 3 denotes the expression of it.

$$f(x) = \max(0, x) \tag{3}$$

3.3.4 Softmax function

It is a combination of multiple sigmoid functions and is basically useful for multi-class classification problems instead of binary classification. It is typically applied at the output layer to categorize the output into one of the predefined classes. The expression for the softmax function is given by Eq. 4.

$$\sigma(z_j) = \frac{e^{z_k}}{\sum_{k=1}^{K} e^{z_k}} \quad for\ j = 1\ldots.K \tag{4}$$

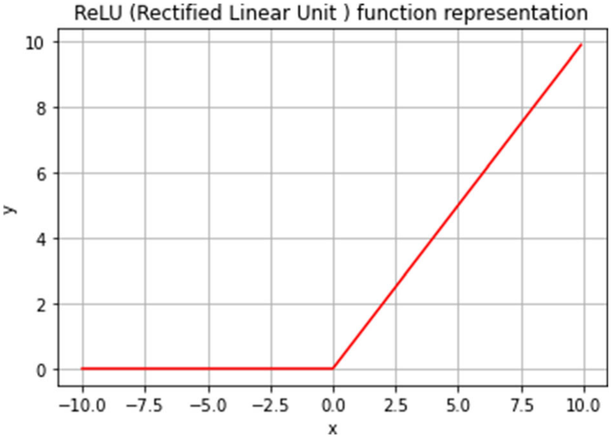

Fig. 7 Graphical representation of ReLU function

3.3.5 Deep learning models

This subsection presents the various deep learning models for HAR.

3.3.6 Deep feed-forward neural networks

Deep Feed Forward Networks are one of the basic Deep Learning models and are also known as MLP (Multilayer Perceptron). They are most suitable for classification problems with supervised classification techniques where the output class labels are already known. The model aims at finding a function f, such that there is a mapping of input x to an output y as y = f(x). It tries to learn the value of the parameters θ which provides the best function approximation in the equation y = f(x;θ).

Figure 8 illustrates a deep feed-forward network model with 3 hidden layers having 100 nodes which are fully connected with ReLU activation function at each layer.

The final fully-connected output layer applies the Softmax function for multi-class classification of different activity classes**.**

3.3.7 Recurrent neural network (RNN)

As opposed to DNN which works on fixed input data length, the Recurrent Neural Network (RNN) has an advantage of working on data sequences having variable input length. It is most suitable for temporal data. It uses the previous layer knowledge to make current predictions. Therefore, it can be termed as having a short term memory unit. There are two approaches for RNNs.

1. **LSTM (Long Short Term Memory)**: It has 3 gates, Input, Output and Forget. The forget gate enables the LSTM to overcome the problem of long-term dependencies by storing only required information and removing unwanted information.

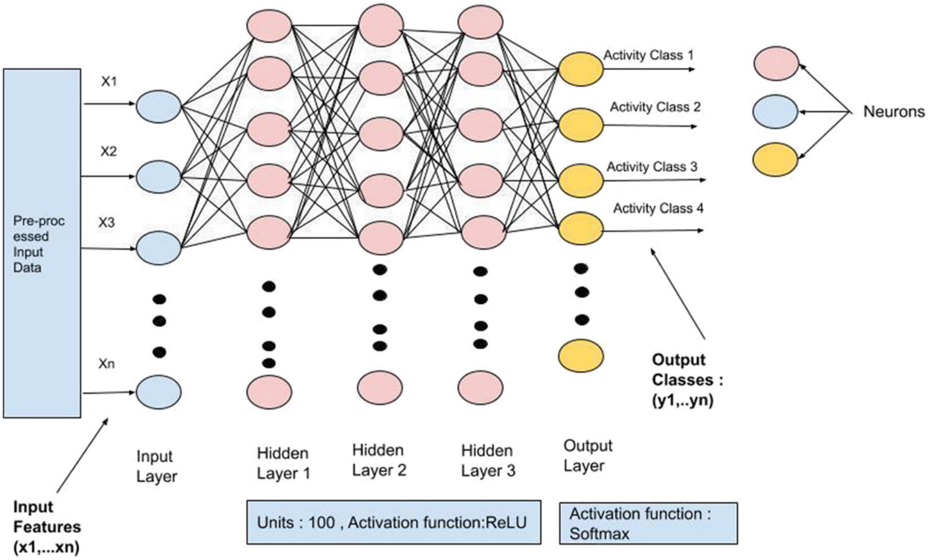

Fig. 8 Deep feed forward network model for HAR

2. **Gated Recurrent Unit (GRU):** It has 2 gates for memory control, input and reset gates. It faces the problem of long-term dependencies as it has to remember data for a large amount of time.

3.3.8 Long short term memory (LSTM)

It is a type of Recurrent Neural Network which also has proven to be helpful in HAR applications. As discussed in [24], the LSTM model tries to map the data segment in each sensor window to the respective activity class, where the observations collected from input sequence are read one at a time. Every time step can contain one or more variables like parallel sequences. The LSTM is beneficial in solving the long-term dependency issue found in Recurrent Neural Networks and also handles the problem of vanishing or exploding gradient with the help of the structure of its cell. The structure of an LSTM cell with its gates is depicted in Fig. 9.

Based on the structure of a LSTM Cell, The equations for LSTM gates are given in (5), (6) and (7) respectively.

$$input_t = \sigma(w_{in}[g_{t-1}, r_t] + b_{in}) \tag{5}$$

$$forget_t = \sigma\left(w_{fo}[g_{t-1}, r_t] + b_{fo}\right) \tag{6}$$

$$output_t = \sigma(w_{out}[g_{t-1}, r_t] + b_{out}) \tag{7}$$

where $input_t$ is the input gate, $forget_t$ is the forget gate and $output_t$ is the output gate.w_r represents the weight value for corresponding gate *(r) neurons.* g_{t-1} indicates the output generated from previous LSTM block at timestamp *t*.Current timestamp is represented as ,r_t, b_r

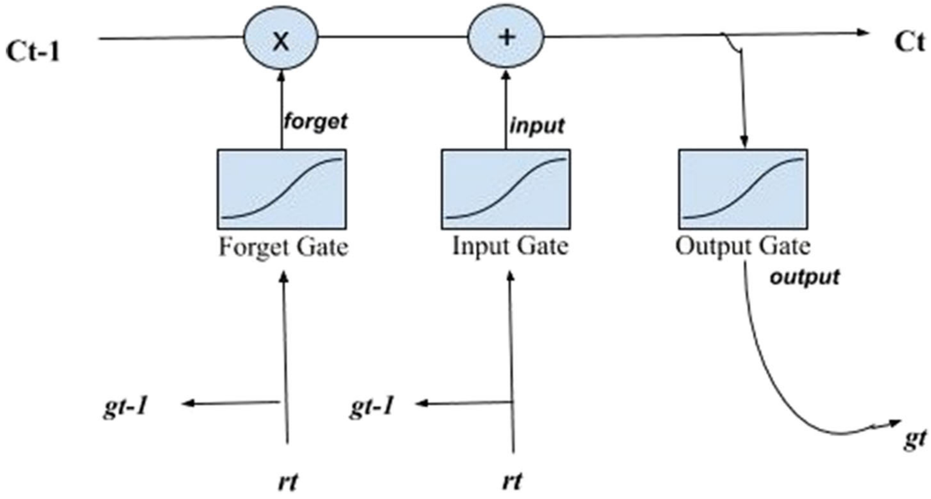

Fig. 9 Structure of a LSTM cell

is the bias value for gate *r* and σ is the activation function used. The advantage of LSTM over CNN is the limitation of CNN that it works on fixed sized window data which LSTM doesn't have. The LSTM predicts corresponding activity class for each input time step with a subsequence of input sensor data, which are aggregated for predicting the resultant activity for each window.

The implementation of a hybrid DNN and GRU based model is illustrated in Fig. 10.

3.3.9 Convolutional neural network

Zeng et al. [29] have proposed a CNN based model for HAR which automatically extracts discriminative patterns while capturing local dependencies of input sensor signal. They applied partial weight sharing to the data to improve the performance. *Sikder* et al. [18] discussed the use of a CNN model for multichannel time–series data for effective HAR. The convolution and pooling layers capture the important features, which are unified among multiple channels and then mapped into corresponding activity. Ijjina et al. [8] have proposed and implemented a model with an ensemble of CNN classifiers for Human activity recognition by changing the initialization of weight values.The output of the trained classifiers was further considered as the confidence level for prediction of specific activity.

Consider $x_j^O = x_{1\dots\dots}\ x_n$represent the input from accelerometer and gyroscope, n is the number of input values per window, then, the initial convolutional layer can be represented as:

$$c_j^{1.k} = \sigma\left(b_k^1 + \sum_{f=1}^{M} w_f^{1.k} x_{j+f-1}^{O.k}\right) \tag{8}$$

Where σ indicates the activation function for layer index 1,b_k is the bias term for kth feature map. The kernel filter size is represented by M, w_f^k is the weight value for *k* feature map with filter index *f.*

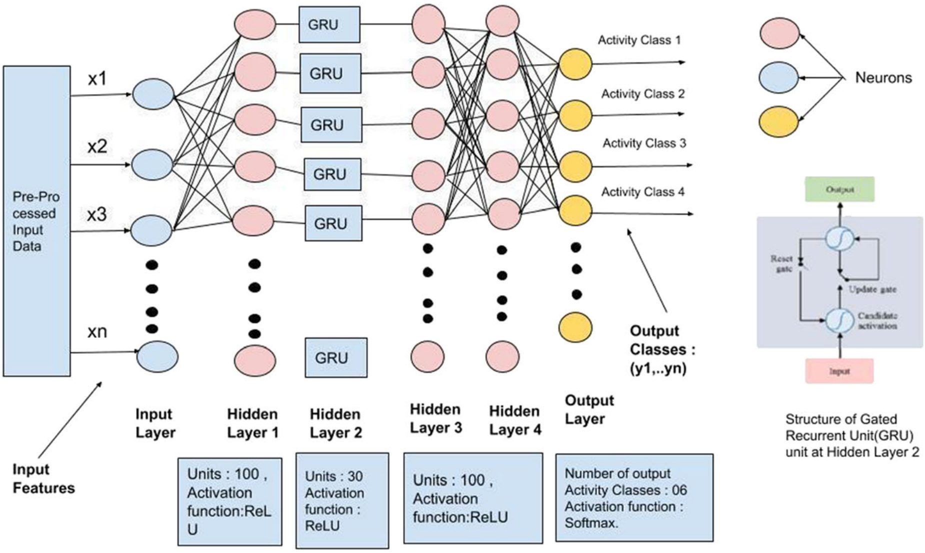
Pre-Pro cessed Input Data
x1
x2
x3
xn
Input Features
GRU
GRU
GRU
GRU
GRU
GRU
Activity Class 1
Activity Class 2
Activity Class 3
Activity Class 4
Output Classes : (y1,..yn)
Neurons
Output
Reset gate
Update gate
Candidate activation
Input
Structure of Gated Recurrent Unit(GRU) unit at Hidden Layer 2
Input Layer
Hidden Layer 1
Hidden Layer 2
Hidden Layer 3
Hidden Layer 4
Output Layer
Units : 100 , Activation function:ReLU
Units : 30 Activation function : ReLU
Units : 100 , Activation function:ReLU
Number of output Activity Classes : 06 Activation function : Softmax.

Fig. 10 A Deep feed forward network + GRU model for HAR

Similarly, based on Eq. (5), the lth convolutional layer output can be identified using Eq. (9).

$$c_j^{l.k} = \sigma\left(b_k^l + \sum_{f=1}^{M} w_f^{l.k} x_{j+f-1}^{l-1.k}\right) \tag{9}$$

3.3.10 Max pooling layer

The maximum value is to be identified from the set of nearby input values after convolutional layer. This task is performed using the max-pooling layer. The value is calculated as:

$$p_j^{l.k} = max_{s\in S}\left(c^{l.k}_{j\times U+s}\right) \tag{10}$$

where, S indicates the pool size and U is the stride value. Multiple stacking of convolution and pooling layers can be used together to build a deep convolutional network.

After the convolution layers perform the feature extraction, the output of the last layer is fed to a fully connected layer which can itself be acting as the layer for classifying activities or there can be a series of fully connected layers with the last layer performing the activity classification using the softmax activation function which is most suitable for multi-class classification problems. For feeding the extracted features from the last convolutional layer to the softmax /fully-connected layer, there is a need of flattening the features in a feature vectors like.

p^l=[p_1....... p_l] where l is the total number of units in the final pooling layer to be fed to a fully connected layer/ softmax layer. Considering the flattened features vectors, The fully connected layer can be represented as:

$$d_j^l = \sum_k w_{ki}^{l-1}\left(\sigma\left(p_j^{l-1}\right) + b_j^{l-1}\right) \tag{11}$$

Where σ is the activity function like ReLU which was used at previous layers of the model, w_{ki}^{l-1} is the weight that connects the layer *l-1 jth* and *kth* nodes and bias is represented as b_j^{l-1} .

The last layer with softmax activation function outputs the corresponding activity class.

$$O(c\mid p) = argmax_{c\in A}\ \frac{\exp\left(p^{L-1}w^L + b^L\right)}{\sum_{i=1}^{N} exp\left(p^{L-1}w_i\right)} \tag{12}$$

Where, L is index of last layer, N represents the total number of classes and c is the activity class. Back-propagation is performed at the fully connected layer by using an Error function with weight adjustment.

The implementation of a CNN model has been carried out using mini-batches of various sizes to compare the performance measures and minimize the negative log likelihood.

The CNN model is depicted in Fig. 11.

The integrated model implemented with the combination of CNN and LSTM for both datasets is depicted in Fig. 12.

The proposed DeepCNNRF Model combines the feature engineering provided by CNN layers. The selected features are fed to the Random Forest, an ensemble learning algorithm which classifies the data into the respective human activities.

3.3.11 Random Forest classifier

Ensemble Learning focuses on combining various classifiers together instead of single classifier, by using voting mechanism for classification. The types of Ensemble Learning algorithms are Bagging and Boosting.

Random Forest is an example of bagging ensemble machine learning which majorly performs the task of classification, regression and feature selection among other tasks. It provides an improvement over the bagging applied to decision trees as seen in CART as discussed by Breiman et al. [25]. It works by having a collection of decision trees, that predict the output activity class individually and a mean of predictions from all the individual trees is used to identify the resultant output activity class. As opposed to CART, Random forests make sure that the individual trees have less correlation. For K input features, a value k, k<<K is selected in such a way that at every node, k variables will be selected randomly out of the input K features and the best possible split on these k variables is used to split the node as discussed in [5]. For classification, the value of k can be selected as the output of *sqrt(K)*.The input feature vector is processed by every individual decision tree and the final classification is performed by using majority voting among the decision trees that are part of the random forest. The overall training time for Random Forest is less and it results in optimal accuracy values. The generalized architecture of the proposed DeepCNN-RF is shown in Fig. 13 and detailed model for HAR is illustrated in Fig. 14.

The proposed architecture includes two-modelled approach where the output of first model is given as input to model 2.The first model provides the extracted features as input to the ensemble of Random Forest for classification purpose. The model has provided performance metrics values nearly to 97% which is more than many existing as well as proposed hybrid deep learning models for human activity recognition applied on the datasets on which the experiments are carried on.

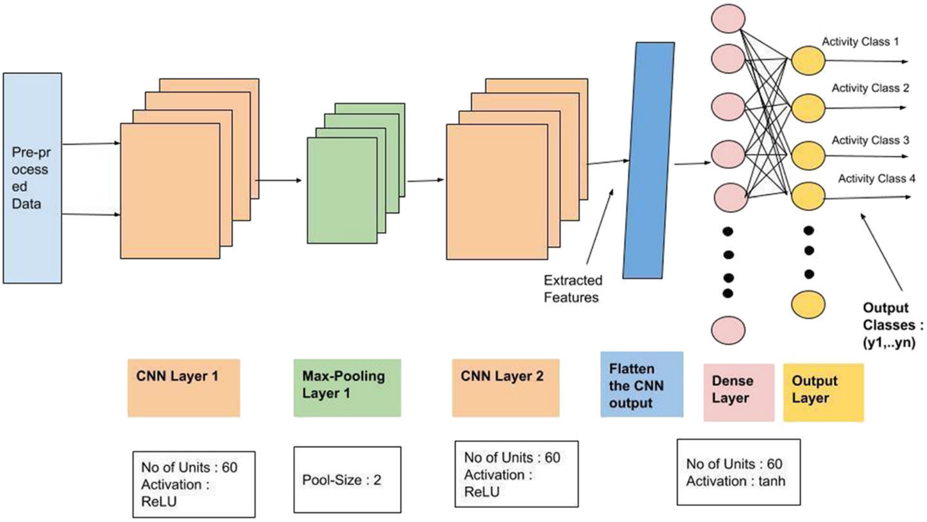

Fig. 11 A deep CNN model

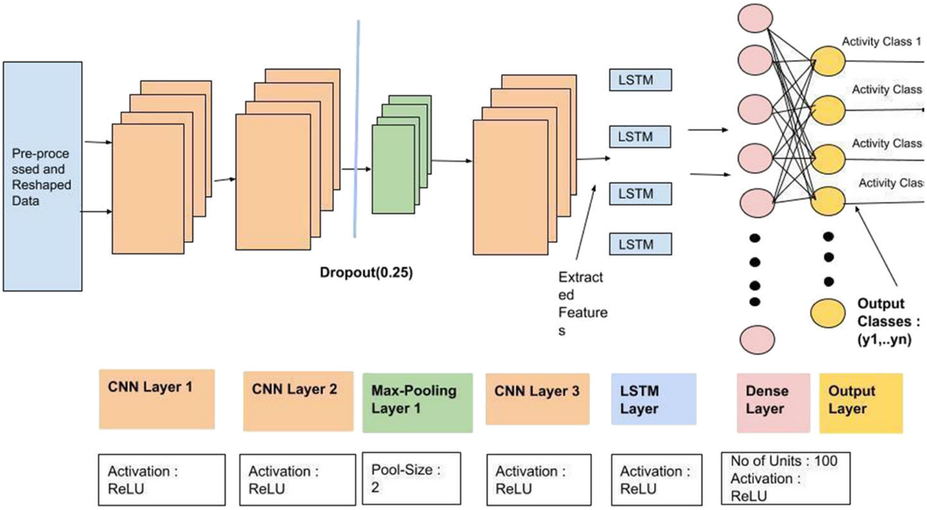

Fig. 12 A deep CNN + LSTM model

3.4 Model compilation and optimization techniques used

Being iterative, Deep Learning models require tuning of various hyper parameters to reduce the model training time and cost function. Thus there is a need of optimization algorithms/ techniques to optimize the value of cost function. The cost function can be mathematically represented as given below:

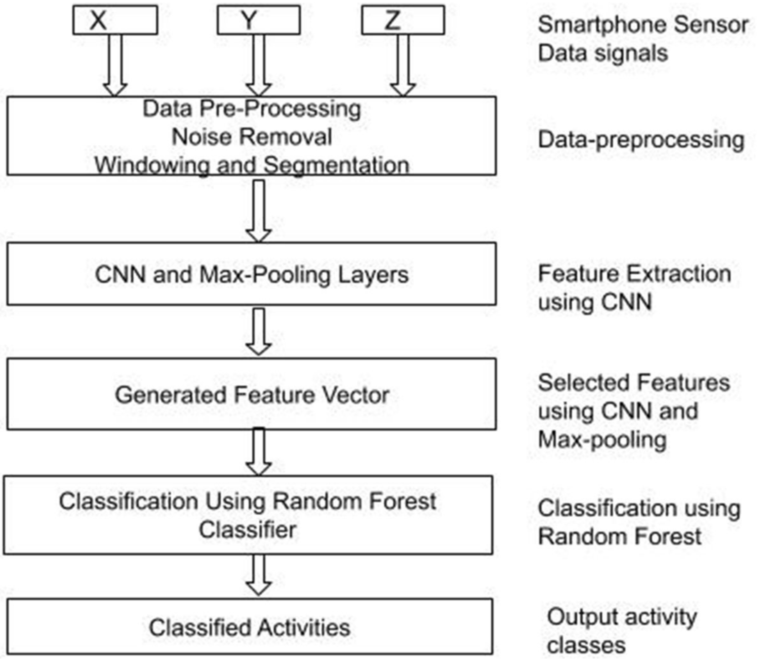
X
Y
Z
Smartphone Sensor Data signals
Data Pre-Processing Noise Removal Windowing and Segmentation
Data-preprocessing
CNN and Max-Pooling Layers
Feature Extraction using CNN
Generated Feature Vector
Selected Features using CNN and Max-pooling
Classification Using Random Forest Classifier
Classification using Random Forest
Classified Activities
Output activity classes

Fig. 13 Generalized architecture of deep CNN-RF model

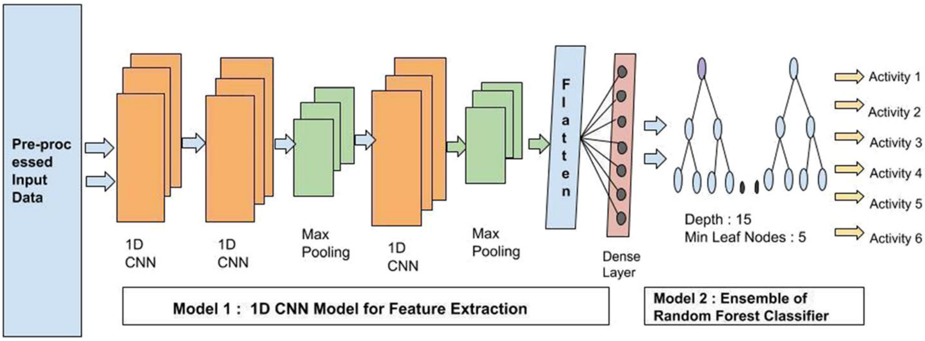

Fig. 14 Proposed deep CNN-RF model for HAR

$$C(W,b) = \frac{1}{m}\sum_{j=1}^{m} L\left(y'^{j}, y^{j}\right) \tag{13}$$

Where C represents the Cost function and can be calculated as the mean of Loss L between the predicted value (y') and actual value(y). y' is obtained while doing forward propagation by taking values of Weight and Bias as W and b respectively. While compiling the model Categorical Cross-Entropy loss function was used as it's the most suitable function for multi-class classification problems. It takes into account the Softmax activation layer and hence is also known as Softmax loss. The loss can be formulated as shown in Eq. 14.

$$CE\ Loss = -\sum_{i}^{C} c_i \, log\left(f(v)_i\right) \tag{14}$$

Where c_i and $f(v)_i$ are the ground truth and Softmax loss for each class i in C classes. There are various optimization techniques which can be used like Gradient Descent, Momentum, RMS Prop and Adam. Among all techniques Adam optimization technique has been used heavily in most of the related research work. It is an acronym for Adaptive Momentum. It combines the advantages of RMS Prop and Momentum techniques together and hence, is one of the most powerful optimization technique used for deep learning.

3.5 Reduction of over-fitting problem

To avoid the problem of overfitting, the Dropout regularization technique, which modifies the network instead of cost function have been applied that allows ignoring of random neurons during the training process. It randomly selects a node to be temporarily removed without affecting the input and output neurons. The dropout rate selected is 0.25 across the layers. To improve the speed and performance of the proposed approaches further, Batch-normalization technique [9] has been applied in few models as it also possesses few regularization properties.

4 Experimental setup and result analysis

4.1 Datasets used for experiments

The proposed models were implemented on WISDM Dataset and UCI HAR which are existing and commonly used datasets for analysis.

Table 1 provides details about the input data from existing HAR Datasets on which the proposed models have been implemented.

For the experiments and model creation, the input data is divided into training and testing sets with a ratio of 70:30.Some part of training data is considered as validation set. For implementing above steps and visualizations Python have been used by importing various essential libraries like Scikit-learn, Keras Version 2.2.5 and Tensor Flow 2 x version as backend for implementing deep learning approaches. For visualization plots, the inbuilt Matplotlib library is imported and used. All the experiments have been performed in Google Collaboratory, provided by Google with required GPU Support essential for deep learning.

4.2 Performance metrics

As the deep learning models are used for classifying the given data into multiple activity classes, various performance measures related to multiclass classification problems have been used and identified. The basic performance measures used for classes *Ci where i* = *1 to t* where tp_i: True Positive, fp_i: False Positive, tn_i: True Negative and fn_i: False Negative.M indicated Macro averaging of performance metrics.The metrics used are calculated as follows:

$$Classification\ Accuracy = \sum_{i=1}^{t} \frac{\boldsymbol{tp_i + tn_i}}{\boldsymbol{tp_i + fn_i + fp_i + tn_i}} \tag{15}$$

Table 1 Details of datasets used

Dataset	Devices and Sensors Used	No of Subjects	Number of data instances	Recognized Activities	Attributes/ Features	References
UCI HAR using Smartphone	Accelerometer and gyroscope	30	10,299	6 Activities Walking Walking Upstairs Walking Downstairs Sitting Standing Lying	561	[28]
WISDM Activity Recognition	Accelerometer	36	1,098,207	6 Activities Walking Walking Upstairs Jogging Walking Downstairs Standing Sitting	6	[2]

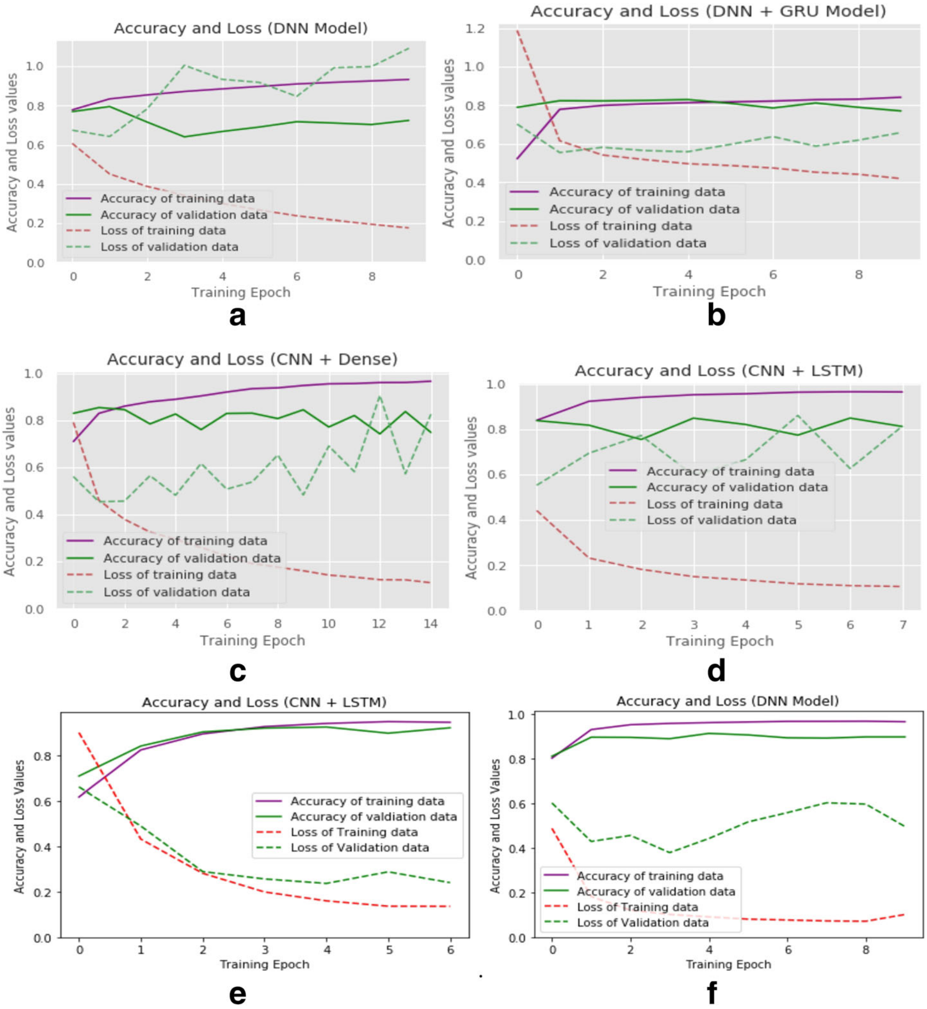

Fig. 15 (**a**–**d**) Accuracy and loss graphs for WISDM dataset. (**e**-**f**) Accuracy and loss graphs for UCI HAR

Table 2 Comparison of accuracy of proposed approaches with existing approaches

Applied Deep Learning Approaches	WISDM Dataset		UCI Dataset	
	Related Work	Our Work	Related Work	Our Work
DNN	–	74%	–	78%
DNN+LSTM		81%		87.79%
DNN+GRU	–	80%	–	90%
CNN	93.32% [7]	88%	95.18% [25]	96%
CNN+LSTM	–	**94%**	–	**97%**
CNN+GRU	–	**82%**	–	96.7
CNN+RF	–	**97.77%**	–	**98.2%**

Table 3 Hyper-parameters tuned for proposed models

Hyper-Parameter	Values taken in to consideration
No of units/nodes (DNN)	100,120
Batch-size	32,128
Kernel size	3,11,16,64
Activation Function(Internal layers)	ReLU(Rectified Linear Unit),tanh
Learning Rate	0.01,0.001
Epochs	5,10,20,30
Dropout	0.25
Pool Size	2
Loss Function	Categorical Cross Entropy
Ensemble(Tree Depth)	15
No of leaf nodes	5

$$Classification\ Precision_M = \frac{\sum_{i=1}^{t} \frac{\boldsymbol{tp_i}}{\boldsymbol{tp_i+fp_i}}}{\boldsymbol{t}} \tag{16}$$

$$Classification\ Recall_M \quad = \frac{\sum_{i=1}^{t} \frac{\boldsymbol{tp_i}}{\boldsymbol{tp_i+fn_i}}}{\boldsymbol{t}} \tag{17}$$

$$Classification\ F1\ Score_M = \frac{\left(\boldsymbol{\beta^2+1}\right)\boldsymbol{Precision_M Recall_M}}{\boldsymbol{\beta^2 Precision_M + Recalll_M}} \tag{18}$$

The training and validation graphs for the datasets with various hybrid models can be seen in Fig. 15 and the comparison of Test Accuracy of proposed and related approaches is provided in Table 2.

It can be observed from Table 2 that the proposed hybrid deep learning approaches by combining various algorithms and also with Random Forest Classifier provides better accuracy than the individual algorithm implementation.

Table 4 Effect of hyper-parameter tuning on accuracy for WISDM dataset

Architecture	Batch Size	No of units	Kernel Size	Epochs	Accuracy (WISDM Dataset)
DNN(3 layers)+Softmax	32	100	–	20	.74
DNN+LSTM	32	128	–	20	.77
DNN+GRU	32	100	–	20	.80
CNN	32	Filters:32	16	5	.81
CNN(3 Layers)+LSTM(30 units)	32		16	15	**.94**
CNN (3 Layers)+GRU(1)	32	Filters: 32	11	10	.81
CNN 3 Layers + Random Forest	128	Filters: 64	**16**	15	**.97**
(Ensemble of RF(15 Depth) (min leaf nodes: 5)	64	Filters: 64	16	10	95.8
	32	Filters: 64	16	**10**	**.97**

4.3 Hyper-parameter tuning and its effect on classification accuracy

Fine tuning of various hyper-parameters has been performed to achieve best possible accuracy values for each approach. Various hyper-parameters and their values taken into consideration for testing purpose are presented in Table 3. The application of parameter tuning of parameters like Filters, Kernel-size for CNN, Number of neurons/units for LSTM and Dense Layer and size of batches, number of epochs for training and their effect on accuracy can be seen in Table 4. It can be seen that with minimum number of epochs and batch-size values like 10 and 32 provided optimal values for various performance metrics including overall accuracy. The hybrid models including CNN have been tuned by changing the filters, kernel size, and number of units etc. to identify the best possible hyper-parameter values. Random Forest Classifier is applied by changing the values of hyper-parameters like depth of the forest as well as the number of leaf nodes etc.

5 Conclusion and future work

This paper proves the advantage of using hybrid deep learning approaches for the emerging field of Human Activity Recognition based on embedded sensors from smartphone like Accelerometer and Gyroscope. Various hybrid deep learning approaches like DNN + GRU, CNN + GRU, CNN + LSTM and CNN + Random Forest have been implemented on two existing datasets UCI HAR and WISDM available for research on HAR. Automated feature extraction using CNN helped to adaptively extract most robust and relevant features and reduce the training time as a whole. The integration of CNN with Random Forest Classifier has added the randomness property while classifying the output classes and therefore improves the overall performance of Human Activity Recognition on the two datasets. In future, the aim will be to focus on improving the performance by taking more number of datasets into consideration and try to achieve transfer learning of proposed approaches for increasing generalization.

References

1. Abbaspour S, Fotouhi F, Sedaghatbaf A, Fotouhi H, Vahabi M, Linden M (2020) A comparative analysis of hybrid deep learning models for human activity recognition. Sensors 20(19):5707
2. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2012) Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In International workshop on ambient assisted living. Springer, Berlin, Heidelberg, pp 216–223)
3. Bayat A, Pomplun M, Tran D (2014) A Study on Human Activity Recognition Using Accelerometer Data from Smartphones. Procedia Comput Sci 34:450–457. https://doi.org/10.1016/j.procs.2014.07.009
4. Bux A, Angelov P, Habib Z (2017) Vision based human activity recognition: a review. In Advances in Computational Intelligence Systems. Cham, Springer, pp 341–371
5. Feng Z, Mo L, Li M (2015) A random forest-based ensemble method for activity recognition. In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, pp 5074–5077
6. Ha S, Yun JM, Choi S (2015) Multi-modal convolutional neural networks for activity recognition. In 2015 IEEE International conference on systems, man, and cybernetics. IEEE, pp 3017–3022
7. Ignatov A (2017) Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. Appl Soft Comput 62:62–922. https://doi.org/10.1016/j.asoc.2017.09.027
8. Ijjina EP, Chalavadi KM (2016) Hybrid deep neural network model for human action recognition. Int J Appl Soft Comput 46:936–952
9. Ioffe S, Szegedy C (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” *ArXiv* abs/1502.03167: n. pag

10. Jiang W, Yin Z (2015) Human activity recognition using wearable sensors by deep convolutional neural networks. In Proceedings of the 23rd ACM international conference on Multimedia, pp 1307–1310
11. Jordao A, Nazare Jr. AC, Sena J, Schwartz WR (2018) Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. arXiv preprint arXiv:1806.05226
12. Ke SR, Thuc HLU, Lee YJ, Hwang J-N, Yoo J-H, Choi K-H (2013) A review on video-based human activity recognition. Computers 2(2):88–131
13. Krishnan N, Cook D (2014) Activity recognition on streaming sensor data. Pervasive Mob Comput 10:138–154
14. Kwapisz JR, Weiss GM, Moore SA (2010) Activity recognition using cell phone accelerometers. Proceedings of the fourth international workshop on knowledge discovery from sensor data (at KDD-10), Washington DC
15. Liu Q, Zhou Z, Shakya S, Uduthalapally P, Qiao M, Sung A (2018) Smartphone sensor-based activity recognition by using machine learning and deep learning algorithms. Int J Machine Learn Comput 8:121–126. https://doi.org/10.18178/ijmlc.2018.8.2.674
16. Mandong A, Munir U (2018) Smartphone based activity recognition using k-nearest neighbor algorithm. In Proceedings of the International Conference on Engineering Technologies, Konya, pp 26–28
17. Mekruksavanich S, Hnoohom N, Jitpattanakul A (2018) Smartwatch-based sitting detection with human activity recognition for office workers syndrome. In 2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON). IEEE, pp 160–164
18. Murad A, Pyun J-Y (2017) Deep recurrent neural networks for human activity recognition. Sensors 17: 2556. https://doi.org/10.3390/s17112556
19. Okai J, Paraschiakos S, Beekman M, Knobbe A, de Sá CR (2019) building robust models for human activity recognition from raw accelerometers data using gated recurrent units and long short term memory neural networks. 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, pp 2486–2491
20. Ordóñez F, Roggen D (2016) Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 16:115. https://doi.org/10.3390/s16010115
21. Ortiz LJ, Olaya AG, Borrajo D (2011) A Dynamic Sliding Window Approach for Activity Recognition. In: Konstan JA, Conejo R, Marzo JL, Oliver N (eds) User Modeling, Adaption and Personalization. UMAP 2011. Lecture notes in computer science, vol 6787. Springer, Berlin
22. Ronao C, Cho S-B (2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244. https://doi.org/10.1016/j.eswa.2016.04.032
23. Sai NL, Samuel K, Naga BK. Vamsidhar E (2019) Performance analysis on human activity detection using knn and random forest. International Journal of Innovative Technology and Exploring Engineering (IJITEE) 8(7):2817–2821
24. Sani S, Wiratunga N, Massie S, Cooper K (2017) kNN Sampling for Personalised Human Activity Recognition. In: Aha D, Lieber J (eds) Case-Based Reasoning Research and Development. ICCBR 2017. Lecture notes in computer science, vol 10339. Springer, Cham
25. Sikder N, Chowdhury Md, Arif A, Nahid A (2019) Human Activity Recognition Using Multichannel Convolutional Neural Network
26. Walse KH, Dharaskar RV, Thakare VM (2016) Pca based optimal ann classifiers for human activity recognition using mobile sensors data. In Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems, vol 1. Springer, Cham, pp 429–436
27. Wang A, Chen G, Yang J, Zhao S, Chang C (2016) A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone. IEEE Sens J 16(11):4566–4578. https://doi.org/10.1109/JSEN.2016.2545708
28. Wang J, Chen Y, Hao S, Peng X, Lisha H (2017) Deep learning for sensor-based activity Recognition: A Survey. Pattern Recognition Lett. https://doi.org/10.1016/j.patrec.2018.02.010
29. Zeng M (2014) convolutional neural networks for human activity recognition using Mobile sensors, 6th International Conference on Mobile Computing. Applications and Services, Austin, pp 197–205. https://doi.org/10.4108/icst.mobicase.2014.257786

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.