Multimedia Tools and Applications (2023) 82:5369–5403
https://doi.org/10.1007/s11042-021-11885-x

1178: PATTERN RECOGNITION FOR ADAPTIVE USER INTERFACES

Inception inspired CNN-GRU hybrid network for human activity recognition

Nidhi Dua[1] · Shiva Nand Singh[1] · Vijay Bhaskar Semwal[2] · Sravan Kumar Challa[1]

Received: 20 February 2021 / Revised: 23 August 2021 / Accepted: 23 December 2021 /
Published online: 9 March 2022

Abstract
Human Activity Recognition (HAR) involves the recognition of human activities using sensor data. Most of the techniques for HAR involve hand-crafted features and hence demand a good amount of human intervention. Moreover, the activity data obtained from sensors are highly imbalanced and hence demand a robust classifier design. In this paper, a novel classifier "ICGNet" is proposed for HAR, which is a hybrid of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU). The CNN block used in the proposed network derives its inspiration from the famous Inception module. It uses multiple-sized convolutional filters simultaneously over the input and thus can capture the information in the data at multiple scales. These multi-sized filters introduced at the same level in the convolution network helps to compute more abstract features for local patches of data. It also makes use of 1×1 convolution to pool the input across channel dimension, and the intuition behind it is that it helps the model extract the valuable information hidden across the channels. The proposed ICGNet leverages the strengths of CNN and GRU and hence can capture local features and long-term dependencies in the multivariate time series data. It is an end-to-end model for HAR that can process raw data captured from wearable sensors without using any manual feature engineering. Integrating the adaptive user interfaces, the proposed HAR system can be applied to Human-Computer Interaction (HCI) fields such as interactive games, robot learning, health

✉ Nidhi Dua
2016rsec001@nitjsr.ac.in

Shiva Nand Singh
snsingh.ece@nitjsr.ac.in

Vijay Bhaskar Semwal
vsemwal@gmail.com

Sravan Kumar Challa
2016rsec002@nitjsr.ac.in

[1] Department of ECE, National Institute of Technology Jamshedpur, Jamshedpur, Jharkhand, India

[2] Department of CSE, Maulana Azad National Institute of Technology, Bhopal, MP, India

monitoring, and pattern-based surveillance. The overall accuracies achieved on two benchmark datasets viz. MHEALTH and PAMAP2 are 99.25% and 97.64%, respectively. The results indicate that the proposed network outperformed the similar architectures proposed for HAR in the literature.

Keywords Convolutional neural network · HAR · Inception · Gated recurrent unit · Wearable sensors · Human-computer interaction · Pattern recognition

1 Introduction

HAR is the process of recognizing various human activities using sensor data. It has gained significant attention from researchers in Human-Computer Interaction and ubiquitous computing [51]. It plays a crucial role in identifying the user interaction with its surroundings and thus can be used as an assistive technology for healthcare, rehabilitation, robotics, and building an intelligent system based on adaptive user interfaces [60, 61]. Furthermore, when Artificial Intelligence (AI), edge computing technology, and adaptive user interfaces are combined, they can be used to create real-time HAR systems [24]. Such HAR systems can analyze and process data within the terminal device where it is collected and helps reduce the latency and provide fast service to the user of the system [27]. Adaptive interfaces can be integrated into the HAR system depending on the specific application. Based on the analysis results, systems with adaptive user interfaces can be used to take appropriate actions, display the results on the screen, or store them in the storage. Such HAR systems can be used in a variety of scenarios, including healthcare (to monitor patients and elders living alone or suffering from diseases like Parkinson's disease, dementia, and others), the smart home environment, tracking and maintaining a healthy lifestyle, video surveillance, postural stability, humanoid robot development, and others [5, 57].

HAR is inherently a pattern recognition task. The HAR framework consists of four steps: data collection, data preprocessing and segmentation, feature extraction, and classification of activities. An overview of the HAR framework is shown in Fig. 1. HAR provides a framework for recognizing human activities by using multimodal data obtained from various sensors. The first step in HAR is to capture the activity data. Various sensors used to capture human activity data mainly include video-based sensors [57], depth sensors [4], wearable sensors [12], smartphone sensors [55], and others.

HAR can be broadly classified into video-based and sensor-based activity recognition. Video-based HAR uses cameras to capture images and videos to recognize human activities

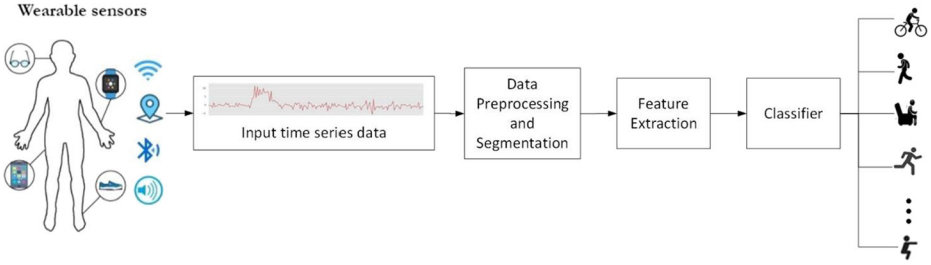


Fig. 1 Overview of HAR framework

and behavior. Computer Vision (CV) based HAR approaches use video sensors to capture activity data. CV-based HAR approaches still suffer from many limitations like - they require proper environmental conditions (like brightness, light, etc.), simultaneous identification and tracking of multiple individuals in an image, occluded targets, and complex background interference. The functioning of CV-based approaches also demands high data processing, hence increases its overall operational cost. Additionally, video-based sensors have restrictions: they can't be mounted/carried everywhere, difficult outdoor installation and maintenance, privacy issues, occlusions, etc. On the other hand, the sensor-based HAR systems are more feasible as they use wearable sensors, smartphone sensors, etc., to capture the activity data. Wearable sensors and smartphone sensors dominate the field of HAR due to their ease of use, low cost, easy installation, and ubiquity. Moreover, wearable sensor based HAR approaches are computationally less expensive in comparison with the CV-based approaches. Sensors like accelerometers, gyroscopes, magnetometers, etc., are widely embedded in devices like smartphones, smartwatches, and armbands that can be easily carried by the user. Consequently, we mainly focused on wearable sensors based HAR in this work.

Wearable and smartphone-based sensors are used to collect activity data. The data so collected are in time-series format. The second step in HAR is data segmentation, and the most common approach to this is using a fixed-length sliding window and splitting the time series data into segments of equal length. The next step is the extraction of features from the data segments obtained in the previous step. Feature extraction is the most crucial step in the process of HAR. Several traditional Machine Learning (ML) based techniques such as Support Vector Machine (SVM), Decision Trees (DT), Random Forest (RF), etc. [2, 9, 31, 38, 42], achieved good performance in inferring human activities. However, these conventional pattern recognition techniques rely largely on hand-crafted features and require domain expertise.

The use of Deep Learning (DL) techniques can simplify the HAR pipeline. DL is a subfield of ML that has achieved excellent empirical performance in several fields like image classification, object detection, image segmentation, image synthesis, etc., [17, 48–50]. DL algorithms have a structure comprising multiple layers of neurons stacked together to extract hierarchical abstractions. Each layer takes its previous feature maps as input and uses a non-linear function to transform them into new feature maps. This hierarchical abstraction allows the DL algorithms to automatically learn the features that best describe the specific application domain. DL employs a deep neural network (DNN) architecture that works on extracting features and classification boundaries by minimizing a certain loss function. DL based techniques don't need any manual feature engineering and can learn the features automatically. CNN, Recurrent Neural Network (RNN), Deep Belief Networks, and autoencoders are DL algorithms abundantly exploited for HAR.

With the emergence of CNNs, the focus in ML research is seen to be shifting to network engineering rather than feature engineering. CNNs are capable of automatic feature extraction and hence don't rely on manual feature extraction. Authors in [54] used a deep CNN to extract features from the segmented frames. They used data obtained from an accelerometer and gyroscope embedded in a smartphone and a multi-class logistic regression classifier for classification. Their method performed inferiorly in detecting stationary activities like 'laying' and 'sitting.'. The authors of [71] proposed converting the data collected by tri-axial accelerometers into a picture format, then identifying human actions using CNN with three convolutional layers and one fully connected layer. In [65], the authors proposed a deep CNN network for feature extraction and classification of activities using activity data acquired from inertial sensors. CNNs perform excellently for local feature extraction within the frame of

data. Still, these approaches do not consider pattern sequences or remember changes in pattern sequences over time, based on the length of gaps between them [43].

RNN's capability to capture temporal context makes it suitable for sequence data [66]. RNN based method proposed in [3] for abnormal behavior and action recognition showed good results but still had room for improvements. The traditional tanh RNN units suffer from vanishing gradient problem and thus lacks in capturing long-term dependencies [7]. But in the case of HAR data, it is necessary to capture long-term dependencies for good classification performance. LSTM and GRU are the RNN variants that can capture long-term dependencies [16] and are thus suitable for HAR. Authors in [26] proposed DNN, CNN, and RNN approaches and performed experiments using three public datasets and concluded that the RNNs outperformed CNNs significantly in detecting short-duration activities having natural ordering, whereas, for prolonged activities like running and walking, they recommended the use of CNNs.

Recognition of daily physical activities is essential to assess the individual's risk of musculoskeletal disorders, diabetes, cardiovascular diseases, and stress [19]. HAR is important in health care to monitor the activities of patients and elders with conditions like dementia, Parkinson's disease, etc. Hence HAR can help detect the abnormalities in regular daily activities and thus preventing any unfavorable consequences. It will be unsafe to compromise in terms of accuracy of HAR systems as the area they are applied to includes healthcare and monitoring of elders. Thus, it is crucial to design HAR systems that can recognize the user activities with accuracy as high as possible. In addition to reasonable accuracy, a HAR model with fewer parameters is desirable for use in real-time embedded applications. So, our primary goal and motivation in this research work are to design a HAR system with fewer parameters that can achieve reasonable accuracy. A system with fewer parameters will have lower computational and memory requirements.

HAR techniques face significant challenges because the raw data from wearable and smartphone-based sensors are largely imbalanced (class-imbalance) [13, 35] and noisy [1]. Moreover, several techniques proposed for HAR depend on heavy data preprocessing and manual feature extraction [2, 31, 42]. Such techniques also demand human intervention and expertise in the field. CNN-based approaches are good at local feature extraction but do not take care of long-term dynamics in the sequence data, whereas modern RNN variants (LSTM and GRU) are good at capturing long-term dependencies. Moreover, in [26], after extensive experimentations (4000 experiments), the authors concluded that the RNNs outperformed CNNs significantly in detecting short-duration activities having natural ordering, whereas, for repetitive and prolonged activities like running and walking, they recommended the use of CNNs.

Thus, to design a HAR system that can overcome the above challenges and achieve the best accuracy when compared to the state-of-the-art HAR techniques, in this proposed work, we choose to combine CNN and RNN to exploit their individual strengths and make the architecture more accurate in recognizing human activities. The RNN variant used for the proposed work is GRU, as it is less complex when compared to LSTM. A DNN based Inception Inspired CNN-GRU hybrid architecture (ICGNet) is proposed in this paper for HAR. It operates on raw sensor data with minimal preprocessing. The network proposed in this paper uses a hybrid of CNN and GRU layers and exploits the advantages offered by both of them. Thus, it is capable of extracting the local features and long-term dependencies in time-series data. The CNN block applied in the ICGNet architecture is inspired by the famous inception-v1 module [58]. It uses filters of different sizes parallelly on the input data and is

thus able to capture multi-scale local features in the data. The CNN block designed for this model uses 1 × 1 convolution [39] to perform channel-wise pooling and thus learns the information across the channels. 1 × 1 convolution is also used to expand and reduce the number of feature maps. For human activity data which is time-series in nature, it becomes imperative to capture the temporal dependencies to recognize the activities precisely. The GRU layers used in the ICGNet architecture enable it to capture the long-term dependencies in the time-series data. Thus, the CNN block and the GRU layers altogether enable the network to capture the diversity of data. The proposed network performance is validated on two publicly accessible datasets viz. MHEALTH [5] and PAMAP2 [52]. The overall accuracies achieved on MHEALTH and PAMAP2 are 99.25% and 97.64%, respectively.

The rest of this paper is arranged as follows: Section 2 presents the non-DL and DL-based methods proposed for HAR in the literature. Section 3 includes dataset description and preprocessing and methodology of the proposed work. Section 4 offers the list of experiments performed, performance metrics used, and the results obtained. The conclusion of this research work is drawn in section 5.

2 Related work

The ubiquitous nature of wearable and smart devices embedded with various inertial and other sensors provides an excellent platform to monitor and infer user activities. In the recent past, numerous works on HAR have explored the task of activity recognition by using data collected by wearable inertial sensors like Accelerometers, Magnetometers, Gyroscopes, and Electrocardiogram (ECG), Heart rate monitor. Many of them used the conventional ML approaches, while more recent ones have effectively exploited various DL-based methods. Some of these HAR-based works using wearable sensor data are discussed in the following sub-sections.

2.1 Non-DL based approaches for HAR

Numerous frameworks for HAR based on Machine Learning (ML) approaches have been proposed. For instance, authors in [35] extracted statistical features (i.e., standard deviation and average) from the raw sensor (accelerometer) data. They analyzed multiple classifiers viz. logistic regression, decision tree (j48), and Multi-Layer Perceptron (MLP); among them, MLP achieved the best results, i.e., 91.7% accuracy value on the WISDM dataset. In [8], the authors designed an ensemble of classifiers, namely J48, MLP, and logistic regression. They validated the ensemble of classifiers on a public dataset comprising of six daily activities. An activity recognition dataset, namely UCI-HAR, was introduced in [2]. The daily activities of 30 subjects were monitored and captured using inertial sensors embedded in a smartphone worn around the subject's waist. Angular velocity and acceleration signals were captured at a 50 Hz sampling rate. These signals, after noise reduction and other preprocessing such as segmentation, go through manual feature extraction. Overall, 561 features (including mean, standard deviation, signal entropy, correlation, frequency signal kurtosis and skewness, and many more) were extracted to describe each activity window. A multi-class SVM made use of these features to classify the activities. In [31], authors combined various features viz. Gaussian Mixture Model (GMM), Electrocardiogram (ECG), the Mel Frequency Cepstral Coefficients (MFCC), and statistical features, and used a Binary Grey Wolf Optimized (BGWO) decision tree classifier. An ensemble approach was proposed in [45]. The authors used an ensemble of

several machine learning techniques viz. SVM, Random Forest, Multilayer Perceptron, Logistic regression, Naive Bayes, and KNN to boost the HAR performance. Jalal et al. in [30] used gyroscope and accelerometer data and preprocessed it using Savitzky–Golay, median and hampel filters. Several features, including binary, wavelet, and statistical features, were extracted, and optimization was done using adam and adadelta. The Maximum Entropy Markov Model (MEMM) was used for the highest entropy. Their technique achieved 90.91% accuracy on the MHEALTH dataset and 91.25% on the USC-HAD dataset. Hidden Markov Models (HMM) were also used for HAR to extract sequential information of the time series data [33]. HMMs are unsupervised approach and hence doesn't require labeled data. But capturing long-term temporal dependencies is difficult for HMMs.

Table 1 lists some of the representative works on HAR using different conventional ML approaches. The ML techniques used for HAR majorly rely on manual feature engineering and heavy data preprocessing. Moreover, they demand domain expertise. Though, the features extracted were heuristic-driven. Still, there were no systematic or common feature extraction methodologies to extract distinguishing traits for human actions successfully. Thus, making the process of HAR complicated.

2.2 DL based approaches for HAR

Several DL-based methods were proposed to overcome the drawbacks and challenges associated with the conventional ML techniques. Table 2 lists some of the representative works on HAR using different DL approaches. DL techniques do not require manual feature engineering; instead, they are capable of automatic feature extraction and don't require advanced domain knowledge. In DL-based techniques, the focus is mainly on designing the network architecture and selecting hyperparameters to obtain optimum performance.

With the advancement in processing capabilities in the recent past, DL-based techniques have seen huge success in various fields like image segmentation, object detection, and several recognition tasks. Several DL-based approaches like CNN, LSTM, CNN-RNN, etc., have been proposed for HAR. Some of the DL based techniques are discussed below:

CNN based methods CNN has seen a surge in various applications area, including HAR. Several CNN-based works have been proposed in the past decade. For instance, in [25], a CNN-based multi-layer network was designed for HAR using accelerometer and gyroscope sensor data. The authors used weight sharing for the CNN models to learn features and classify the multimodal data. In [14], a CNN model using conditionally parameterized convolutions was proposed for HAR in real-time. In [29], yet another CNN-based technique was proposed for HAR using accelerometer data. It used CNN for local feature extraction and extracted statistical features to capture the global characteristics of the time-series signal. The use of statistical features makes this technique fall in the hand-crafted feature engineering approach. A deep CNN architecture was proposed in [40] for the classification of multivariate time-series data. The authors designed a novel scheme for input tensor transformation and the local interactions among the variables were captured by the convolution operation. Another CNN model was designed in [63] for HAR using the data obtained from smartphone sensors. In [61], the authors extracted features using Gaussian kernel-based Principal Component Analysis (PCA) and then trained a CNN classifier using the extracted features. In [15], a divide and conquer based 1D CNN was proposed. Six different activity classes were first divided into two groups of static activity class and dynamic activity class. Then within each class again, a model

Table 1 Representative works in HAR using ML techniques. (A = Accuracy, F=F1-score)

Author, Year	Method	Strength	Limitations	Evaluation metric A/F (Dataset)
Jennifer et al. [35], 2010	Accelerometer data were used and extracted several statistical features. Evaluated three different classifiers viz. J48, LR, and MLP.	MLP classifier achieved good accuracy	Manual feature engineering	A=91.7% (WISDM)
Lara et al. [36], 2012	Accelerometer & Vital sign data were used, and extracted several statistical and structural features. Eight different classifiers were evaluated	Good accuracy achieved.	Hand engineered feature extraction	A=95.7%
Anguita et al. [2], 2013	UCI-HAR dataset for HAR was introduced. Various time and frequency domain features were extracted, and SVM based classifier was used to infer the activities.	Good accuracy achieved on UCI-HAR dataset	561 features were extracted per window.	A=96% (UCI-HAR)
Cagatay et al. [8], 2015	Inertial sensors data were used. Ensemble of DT, MLP, and LR classifiers was used by the voting mechanism.	Better performance than standalone MLP based HAR	Increased complexity due to the use of an ensemble of classifiers	A=91.61% (WISDM)
Jalal et al. [31], 2020	Accelerometer, Magnetometer & Gyroscope data were used, and a Notch filter was applied for preprocessing. Features extracted include MFCC, ECG, GMM, and various statistical features. BGWO optimized DT classifier was used.	Varied features extracted help to recognize the activities with reasonable accuracy.	Complexity was increased due to the huge number of features extracted.	A=93.95% (MHEALTH) A=88.25 (MOTIONSENSE) A=96.83% (IM-AccGyro)
Nguyen et al. [45], 2019	Various time and frequency domain features were extracted. An ensemble of multiple classifiers viz. LR, MLP, SVM, KNN, NB, and RF was proposed using the voting rule.	Robust classifier compared to previous works.	The complexity of the model was increased due to feature extraction and ensemble of six base classifiers	A=94.72% (MHEALTH) R=83.20% (USCHAD)
Jalal, et al. [30], 2020	Gyroscopes & Accelerometers data were used and preprocessed using Savitzky–Golay, median and hampel filters. Extracted binary, wavelet, and statistical features. Used Adam and Adadelta for feature optimization, MEMM was used for highest entropy.	Decent detection performance achieved.	Increased complexity due to a large number of features used. Heuristic driven process.	A=90.91% (MHEALTH) A=91.25% (USC-HAD)

was used to classify individual activities. A minimum of three recognition models was required in the proposed two-stage HAR, thus increasing the overall model complexity. The CNN-based approaches can extract translational local features but fail to capture global temporal dependencies in sensor data [47]. Thus, in this aspect, CNN alone architectures fall short in the case of HAR, where capturing global dependencies in the time series activity data is crucial to precisely detect the activities.

RNN based methods RNN is another DL-based algorithm extensively used for activity recognition. HAR data obtained from wearable sensors are in the form of time series, and hence the temporal dependencies need to be captured. RNNs are thus widely employed in HAR as they are designed to capture temporal dependencies [66]. In [10], the authors proposed an LSTM based network for HAR which was comprised of two LSTM layers. The model was validated only on a single dataset containing six daily activities and achieved 92.1% accuracy. A residual-bidirectional LSTM based network was designed in [70] for HAR using data acquired by smartphone sensors. The residual connection used in the architecture helped the model to converge faster, and the architecture achieved an F1-score of 90.5% on the OPPORTUNITY dataset. In [67], a bi-dir LSTM network for human activity recognition was proposed and used the data obtained from the gyroscope and accelerometer sensors embedded in a smartphone. The network used raw sensor data from the UCI-HAR dataset and obtained an accuracy of 93.79%; however, the method was validated only on a single dataset. Authors in [62] designed an LSTM based network. The data obtained from the sensors were first normalized and then fed to a stacked LSTM network and a softmax classifier. The model was validated only on a single dataset (UCI-HAR) and achieved an accuracy of 93.13%. An attention-based LSTM model was proposed in [69] for HAR. The authors applied temporal attention to the hidden layer of LSTM and sensor attention to the input layer of the LSTM. They also applied continuous attention constraints on both types of attention and validated the model on three datasets. However, their method could achieve accuracy values below 90% on all the three datasets used. Hammerla et al. [26] proposed deep, convolutional, and recurrent models for HAR. They performed around 4000 experiments and established benchmark results on three datasets. The LSTM and Bi-LSTM network they proposed achieved an F1-score of 91.2% and 92.7% on the OPPORTUNITY dataset and an F1-score of 88.2% and 86.8% on the PAMAP2 dataset. The authors also concluded that RNN networks outperformed CNNs in the detection of activities that have natural ordering but are short in duration, whereas CNNs outperform RNN in the detection of activities that are repetitive and prolonged, like walking and running. So, several approaches for HAR designed the network by combining both CNN and RNN to take advantage of both in the same network. Some of these CNN-RNN hybrid works are discussed in the following subsection.

CNN-RNN hybrid methods Several applications like residential load forecasting, soil moisture prediction, digit recognition, etc. [46, 56, 68], in addition to HAR, benefit by using a combination of the CNN and RNN networks. Various recent approaches for HAR proposed network architectures comprising both convolutional layers as well as recurrent layers. For instance, in [12], authors proposed a CNN-LSTM hybrid approach called 'DEBONAIR' to recognize complex human activities. They designed separate convolutional subnetworks to capture features from various sensor signals. Their model achieved an F1-score of 83.6% in detecting seven complex activities from the PAMAP2 dataset. The network proposed in [64] comprised two LSTM layers succeeded by convolutional layers, followed by a GAP layer, a

Table 2 Representative works in HAR using DL techniques. (A = Accuracy, F=F1-score)

Author, Year	Method	Strength	Limitations	Evaluation metric A/F (Dataset)
Ronao and Cho [54], (2016)	1D CNN was used to extract transitional invariant and hierarchical features from accelerometer and gyroscope sensor data, and a multi-class logistic regression classifier was used for classification.	Good classification accuracies for classification of moving activities like 'walking downstairs' and 'walking upstairs.'	Performance was inferior in detecting stationary activities like 'laying' and 'sitting.' This method failed to capture temporal variance.	A=94.79% (UCI-HAR)
Hammerela et al. [26], 2016	Multiple DL-based models were presented. Performed extensive experiments and validated the model on three datasets.	The LSTM and BiLSTM based models outperformed CNN models in the detection of activities that are of short duration.	They observed that LSTM and BiLSTM models underperformed in the detection of activities that were repetitive and of longer duration.	*LSTM model* F1=91.2% (OPPORTUNITY) F1=88.2% (PAMAP2) *BiLSTM model* F1=92.7% (OPPORTUNITY) F1=86.8% (PAMAP2)
Ha et al. [30], 2016	2D-CNN with partial and full weight sharing	Both common and modality-specific characteristics were captured.	CNN could extract translational local features but failed to capture global temporal dependencies in sensor data.	A=91.94% (mHealth)
Lingjuan et al. [41], 2017	A hybrid of LSTM and CNN was proposed.	The privacy-preserving technique maintains the privacy of the user data.	Dataset was randomly divided as 70–15-15 for training, validation, and testing, respectively. Random partition of data is not desirable in the case of HAR.	A=96.2% (MHEALTH) A=98% (UCI-HAR)
Ignatov et al. [29], 2018	Ensemble approach combining features extracted by CNN and Statistical features.	CNN extracts local features, and statistical features capture the global information in time series data	Dependency on statistical features. Feature extraction was not completely automatic.	A=97.6% (UCI) A=93.32% (WISDM)
Cho and Yoon [15], (2018)	Divide and conquer based 1D CNN was proposed. Six different activity classes were first divided into two groups of static activity class and dynamic activity class. Then within each class again, a model was used to classify individual activities.	The accuracy achieved was high when compared to a single 1D CNN end-to-end model.	Ineffective when the number of activities is increased. Overall model complexity was high when compared to a single 1D end-to-end CNN model.	A=97.62 (UCI-HAR) A=94.27 (OPPORTUNITY)

Table 2 (continued)

Author, Year	Method	Strength	Limitations	Evaluation metric A/F (Dataset)
Zhao et al. [70], 2018	A residual-bidirectional LSTM based network was designed.	Residual connection in the architecture helped the model to converge faster	Activities with static behavior were sometimes misrecognized.	F1=90.5% (OPPORTUNITY) F1=93.6% (UCI-HAR)
Zeng et al. [69], 2018	LSTM with Continuous temporal and Continuous sensor attention networks were used.	Able to focus on chief sensor modalities and salient signal components. Better performance compared to baseline RNN approach.	Recognition performance achieved is still low.	A=89.96% (PAMAP2) A=83.73% (Daphnet Gait) A=89.03% (Skoda)
Uddin et al. [61], 2019	Gaussian kernel PCA was used for feature extraction and Z-score normalization to normalize the features obtained. Deep CNN was trained using obtained features and is used to classify the activities.	Superior performance compared to ANN and DBN	Manual feature extraction.	A=93.90% (MHealth)
Chen et al. [11], 2019	A recurrent Convolutional Attention model was proposed.	A semi-supervised approach thus requires less amount labeled data.	The number of parameters was high. More training and test time were required.	A=94.05% (MHEALTH) A=83.42% (PAMAP2) A=81.32% (UCI-HAR)
Chen et al. [12] (2020)	They used specific convolutional subnetworks to extract features from a different sensor signal.	Specific subnetworks for different sensors data were found to be effective	Low performance on complex human activities	F1=83.6% (PAMAP2)
Wan et al. [63], 2020	A deep CNN model was proposed.	The model was validated on two public HAR datasets, and decent performance was achieved.	Extraction of long-term dependencies not taken care of. Still, there is scope for performance improvement. Data split was done randomly and not on the basis of user-id.	A=92.71% (UCI) A=91% (PAMAP2)
Nidhi et al. [20], 2021	CNN-GRU model. Used multiple sized kernel in CNN layers	Conducted experiments on multiple HAR datasets and achieved decent performance.	Complex architecture and total parameter count are high.	A=96.2% (UCI-HAR) A=97.21% (WISDM) A=95.27% (PAMAP2)
Ronald et al. [53], 2021				A=95.09%

Table 2 (continued)

Author, Year	Method	Strength	Limitations	Evaluation metric A/F (Dataset)
	An inception-based CNN network was proposed.	Easily scalable. It was validated on four datasets.	The no. of parameters and the training time required was very high.	(UCI-HAR) A=88.14% (Opportunity) A=93.52% (Daphnet) A=89.09% (PAMAP2)

BN layer, and a softmax layer. In [44], a CNN-LSTM based HAR classifier was designed. It takes input data from multimodal sensors and performs the classification. Experiments were carried out using the gyroscope and accelerometer data. Another approach for HAR [41] proposed an LSTM-CNN architecture that achieved an accuracy of 96.2 and 98% on MHEALTH and UCI-HAR datasets. The model presented in [20] used a hybrid of CNN and GRU layers. The convolutional layers used three different filter sizes at both the convolutional layers. The authors validated the model on three public HAR datasets viz. PAMAP2, WISDM, and UCI and achieved accuracy values of 95.27, 97.21, and 96.20%, respectively. The model showed good detection performance; however, its architecture is complex, and the number of parameters is high. All these works demonstrated that the network formed by the combination of convolutional and recurrent layers could achieve higher accuracies when compared to CNN architectures. Thus, in this work, we have designed a network that uses both CNN and GRU layers.

Inception based methods With the advent of 'AlexNet' [34], CNNs have become the most used architectures to learn and extract features from the input data to distinguish one category from others. Szegedy et al., by introducing 'GoogleNet' [58], took CNNs one step further in extracting distinguishing features from the data while taking down the total number of model parameters yet achieving higher accuracy. The key component of the Inception-v1 architecture is the inception module, which comprises four branches, as shown in Fig. 6a: a 1 × 1 Conv branch, 1 × 1 followed by 3 × 3 Conv branch, 1 × 1 followed by 5 × 5 Conv branch, and a max-pooling branch. The 1 × 1 convolutions have two purposes, firstly to extract cross-channel correlations, and secondly, to reduce (or sometimes increase or maintain the same) the number of channels in the input (or input feature map) [39]. Inception-based methods used a strategy of split-transform-merge, i.e., the input is split into few lower-dimensional features by using 1 × 1 convolution. These features are then transformed by a set of transformations using multiple-sized kernels (3 × 3, 5 × 5, etc.) and then finally concatenated. This strategy helps them achieve higher accuracy without increasing the complexity compared to the deeper architectures [59].

There are few other inception-based models proposed in recent researches for multivariate time series classification. For instance, [22] proposed domain agnostic architecture "InceptionTime" for the classification of multivariate time series data. It consisted of two residual blocks, where each block contained three inception modules. Each block's input is transferred to the next block's input via a shortcut connection, thus avoiding the problem of vanishing gradients. The residual blocks were followed by a Global Average Pooling (GAP) layer, and finally, the softmax layer classifies the input. The InceptionTime architecture could achieve good classification performance over the UCI archive datasets. However, it wasn't validated specifically on any HAR datasets. Another inception-based architecture called "iSPLInception" was proposed in [53]. The authors designed this architecture for HAR and validated their model on four public datasets. The shorthand notation for iSPLInception architecture can be written as Input layer - > BN layer - > modified inception module (using 1 × 1, 1 × 3, 1 × 5 1D convolution and 1 × 3 1D pooling) - > BN - > ReLU - > residual connections (connecting the input of previous layer to the next inception module) - > ReLU - > GAP1D (1 dimensional GAP layer) - > Softmax layer. The number of inception modules to be used is scalable and depends on the depth parameter. The model achieved an accuracy of 89.09% on the PAMAP2 dataset. The architecture was complex, and the number of parameters was very high.

From the literature, we have made some observations. Firstly, CNNs are good at the extraction of local features but couldn't capture global characteristics of the sequence data. Secondly, the RNN variants like LSTM and GRU can capture the temporal context in the time-series data. Thirdly, the multi-branch inception-based architectures could capture the diversity of information in the data by using a set of transformations by applying different filter sizes (3 × 3, 5 × 5, etc.) and then concatenating the feature maps obtained from all the branches. This strategy helps the architecture achieve higher accuracy without increasing the computational requirements compared to the deeper architectures. Lastly, 1 × 1 convolutions help extract the cross-channel correlations and can also be used to scale the channel dimension. All these observations help us develop the ICGNet model for HAR. In the proposed ICGNet model, we have exploited the strengths of both convolutional and recurrent networks. This model consists of a CNN and an RNN Block. The CNN block is designed by taking inspiration from the inception_v1 module. As shown in Fig. 6b, the original inception module is modified and is then used as the CNN block. The CNN block used in ICGNet significantly differs in structure from the original inception module. The RNN variant used in the ICGNet model is GRU. The detailed architecture of the proposed ICGNet is discussed in Section 3.

2.3 Contributions

The main contributions of this research are as below:

1. A CNN-GRU hybrid network is designed for HAR using data from wearable sensors. CNN is good at local feature extraction, and GRU well captures the long-term dependencies. Hence the hybrid network can capture the diversity of information within the data.
2. The CNN block is designed using multiple filter sizes applied over the input at the same level. It is thus able to capture multi-scale information within the current segment of the data. In addition to multi-sized filters, the CNN block also exploits the strength of 1 × 1 convolution operation to pool the information across the channels.
3. The proposed ICGNet uses raw sensor data with nominal preprocessing and does automatic feature extraction without requiring any expert intervention.
4. The ICGNet model is validated on two benchmark datasets viz. MHEALTH and PAMAP2. The overall accuracies achieved on MHEALTH and PAMAP2 are 99.25% and 97.64%, respectively. The results show that the proposed model outperformed other state-of-the-art HAR techniques using data from wearable sensors.
5. To validate the proposed approach, some standard benchmark deep learning models (deep CNN, stacked LSTM, CNN-LSTM, CNN-GRU, and the inception-based iSPLInception) from the literature have been considered and implemented with the standard public datasets (PAMAP2 and MHEALTH). The performance of the models has been evaluated using standard evaluation measures (Recall, Precision, F1-score, and Accuracy), and the proposed ICGNet model outperformed all the benchmark models. We advocate the usage of ICGNet as the optimal model for HAR.
6. The results of experiments indicate that the proposed model is less complex in terms of the total number of parameters used compared with other state-of-the-art techniques. A lightweight model is desirable for real-time embedded implementation and thus making our model suitable for the purpose.

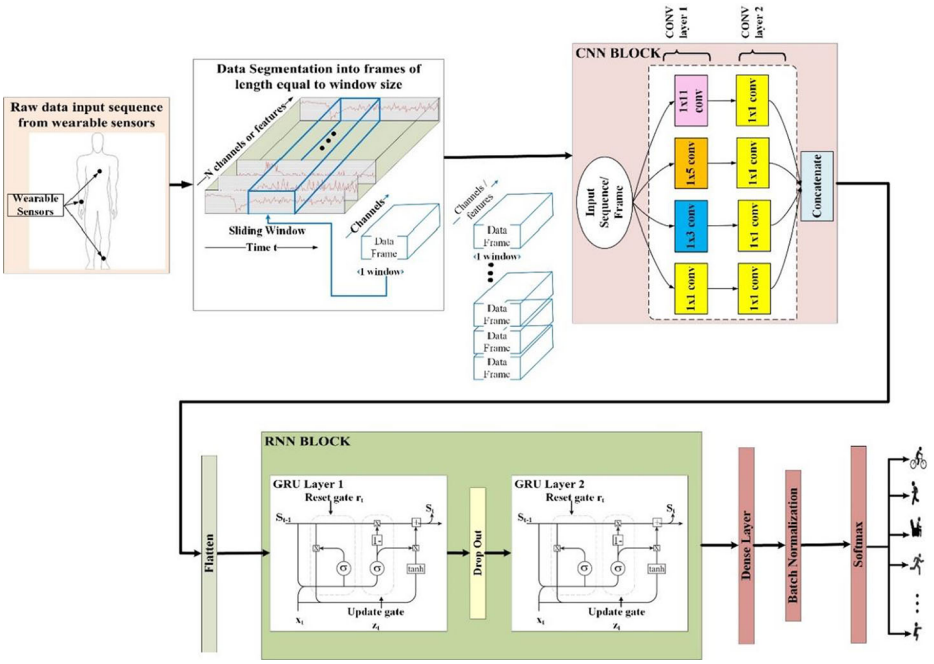


Fig. 2 Block diagram of the proposed HAR framework

3 Materials and methods

HAR is essentially a pattern recognition problem, which comprises steps like data preprocessing and segmentation, feature extraction, and finally, classification of activities. Figure 2 depicts the complete process flow followed in this paper. The first block shows the capturing of activity data through wearable body sensors. The captured human activity sequence data is in time-series format and is segmented using the sliding window technique. The segmented data frames are then forwarded to the CNN and RNN blocks for feature extraction, and finally,

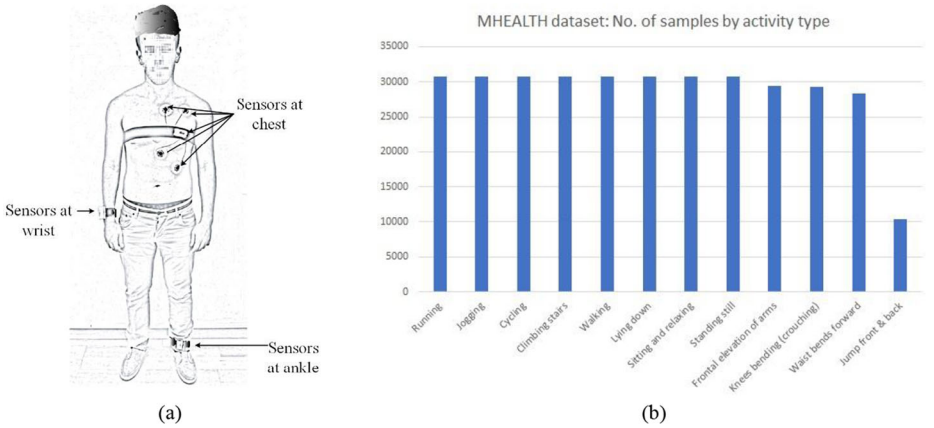


Fig. 3 **a** Placement of wearable sensors on the subject's body. **b** Distribution of activity instances by the type of activity for the MHEALTH dataset

the dense classifier layer with SoftMax activation classifies the data. Each block of the proposed HAR framework is explained in the following subsections.

3.1 Datasets used and data preprocessing

The activity data collected by wearable sensors are in the form of a time series. The extraction of temporal features is essential to recognize the basic actions and the changeovers in the activities. The raw sensor data are the input for the activity recognition task, and the output is the activity class. The datasets used in this paper for experiments are MHEALTH and PAMAP2.

MHEALTH The MHEALTH (Mobile Health) dataset is made available by the UCI repository, and it consists of data of 12 activities performed by ten subjects. The activities recorded in this dataset are climbing stairs, cycling, frontal elevation of arms, jogging, jump front & back, knees bending (crouching), lying down, running, sitting and relaxing, standing still, waist bends forward, and walking. The data were recorded using sensors placed at the chest, right wrist, and left ankle. Figure 3a shows the placement of sensors. The sensors used for experiments were accelerometer, magnetometer, and gyroscope. The attributes recorded by the accelerometer, gyroscope, and magnetometer captured in all three x, y, and z-direction are (a_x, a_y, a_z), (g_x, g_y, g_z), and (m_x, m_y, m_z), respectively. The use of multiple sensors helps measure the motion experienced by different body parts, such as the rate of turn, acceleration, and direction of the magnetic field. ECG measurements can be used for basic health monitoring, monitoring the effect of various activities, and etc. Besides, the electrocardiogram (ECG) signals were also recorded. Two attributes correspond to the ECG lead1 and lead2 signals. At the chest, accelerometer and ECG signals were recorded; at the right wrist and left ankle, accelerometer, gyroscope, and magnetometer signals were recorded. So, a total of 23 attributes/features were captured, comprising all three locations. All sensing modalities were recorded at a sampling rate of 50 Hz. In this paper, data from 8 users are used for training, and that from the remaining two users are used for testing. Figure 3b depicts the distribution of samples by the type of activity.

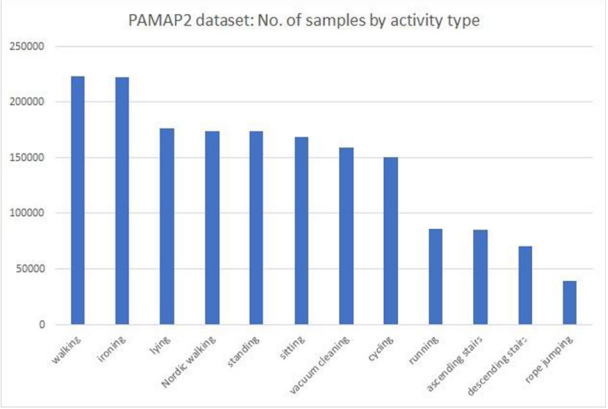


Fig. 4 Distribution of activity instances by the type of activity for the PAMAP2 dataset

PAMAP2 This dataset comprises a total of 18 daily activities recorded for nine subjects. Out of 18 activities, 6 are optional activities (like folding laundry, watching TV, etc.), and the other 12 are protocol activities like (running, rope jumping, cycling, etc.). The distribution of samples by the type of activity performed is depicted in Fig. 4, which indicates that the PAMAP2 dataset has a class imbalance. The actions were recorded using three Inertial Measurement Units (IMUs) and a heart rate monitor. The IMU carries 3-axis sensors to measure acceleration, angular rate, magnetic field, and one temperature sensor. The three IMUs were worn by subjects, one each at the chest, the wrist of the dominant arm, and on the dominant side's ankle. The signals from IMU were sampled at 100 Hz, and the data from IMU was sampled at 9 Hz. Each IMU records 17 features (temperature data, 3D acceleration data, 3D gyroscope data, 3D magnetometer data, and orientation data). The dataset is comprised of a total of 52 features. For this research work, data of two subjects are used for testing, and data of the remaining seven subjects are used for training. In this research work, a total of 21 features captured using the accelerometer (a_x, a_y, a_z), gyroscope (g_x, g_y, g_z), and temperature sensor placed at the chest, hand, and ankle are used for the experiments.

The initial step in time-series data classification is to segment the data into fixed-size frames. The segmentation of time-series data is shown in Fig. 2. Using the sliding window technique, the raw data collected by wearable sensors is segmented into frames. The window size selected for the proposed architecture is 256, i.e., each segment of data (or frame) will contain 256 timestamps per frame and 'n' features (or channels) associated with each timestamp, here n = 23 for the MHEALTH dataset and n = 21 for the PAMAP2 dataset. Hence the size of the input vector is (256, n). For this research work, the data are normalized to have values between 0 and 1.

3.2 CNN-GRU hybrid

DNNs are capable of extracting the features automatically without needing any expert intervention. Hence, the use of DNN as a feature extractor helps build an end-to-end model capable of handling everything from feature extraction to classification. The ICGNet architecture proposed in this paper is depicted in Fig. 8. The proposed network is a hybridization of CNN and RNN layers. The proposed ICGNet architecture consists of both convolutional layers and GRU layers, hence can be called as a CNN-GRU hybrid network. The below subsections explain briefly how the proposed ICGNet uses the strengths of CNN, Inception module, and RNN to extract features from the sensor data.

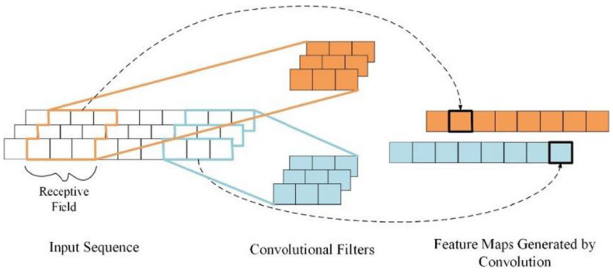


Fig. 5 Convolution operation on 1D input sequence data

3.2.1 Inception module based CNN block

CNNs are widely used in multiple tasks such as image classification, time series forecasting, etc., and provide decent performance due to their weight-sharing concept [37]. The convolution operation on 1D sequence data is shown in Fig. 5. The convolution layer is made up of a set of filters (or kernels). This set of filters are applied to the input signal in a sliding window fashion. Each filter is a matrix of integers that is applied on a subset of the input values of the same size as that of the filter. This subset of the input is known as the receptive field of the filter. Hence, the filter is said to have a local receptive field. The receptive field's values are multiplied by the filter's corresponding values. All the values thus obtained are summated to obtain a single value of a feature map. The filter is slid over the complete input, and the convolution operation is performed at each position the filter is applied over the input. Thus, the convolution layer's output is the multichannel feature maps, where the number of channels in the feature maps is equal to the number of filters in the convolution layer.

The images or speech signals have a strong 2D structure, whereas time series data possess a strong one-dimensional structure, i.e., the spatially or temporally close variables are strongly correlated [37]. Extraction of local features is important to capture the local correlations. CNN can capture these correlations and hence can extract local features by the property of local receptive fields [37].

The proposed model's convolution block is inspired by the inception module introduced in [58]. However, the CNN block designed for ICGNet is not entirely similar to the original inception module. Figure 6a depicts the structure of the inception module. The inception module comprises four branches, a 1 × 1 Conv branch, a 1 × 1 followed by a 3 × 3 Conv branch, a 1 × 1 followed by a 5 × 5 Conv branch, and a max-pooling branch. This module made use of two key concepts. First, it used the idea of multiple-sized filters applied simultaneously over the input. Different sizes of filters have different local receptive fields, and thus, these multiple-sized filters will help compute more abstract features for local patches of data [58].

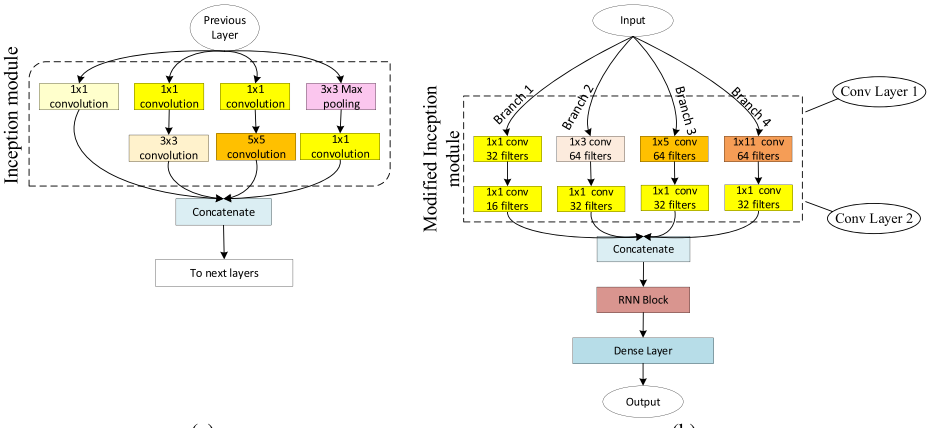


Fig. 6 Difference in the structure of CNN block of the proposed model and the Inception module **a** Inception Module CNN block **b** Modified Inception module used as CNN Block in ICGNet

The second key concept used in the inception module is 1 × 1 convolution. 1 × 1 convolution was first proposed in [39] and as a specific implementation of cross-channel parametric pooling, which enables learning across the channels. 1 × 1 convolution provides channel-wise pooling rather than average or max-pooling across width/height (in case of image data) or length (in case of 1D time-series data). In the inception module, 1 × 1 convolution filters were also used for dimension reduction and hence save the computational requirement.

In this work, the inception module is modified and used as the CNN block of the proposed ICGNet. It is depicted in Fig. 6b. The CNN block of the ICGNet model consists of multiple-sized filters applied parallelly over the input data. The modified inception module employs filter sizes of 1, 3, 5, and 11. Different filter sizes applied parallelly across the input enable the CNN to capture information at diverse scales because different filter sizes will have different-sized receptive fields. Time-series data exhibits one dimension less when compared to image data. In this paper, for 1D time-series data, the convolution operation using a filter size of 1 will be referred to as 1 × 1 convolution. The modified inception module comprises four branches, viz. 1 × 1 followed by 1 × 1 Conv, 1 × 3 Conv followed by 1 × 1 Conv, 1 × 5 Conv followed by 1 × 1 Conv, and 1 × 11 Conv followed by 1 × 1 Conv. The input to the module is the 1D time-series data segmented into frames. Each frame is of the length of window size (used for segmentation), i.e., 256, and of depth/channels equal to the number of features in the input data. The input is passed through the convolutional layer 1, and the generated feature maps are then forwarded to the convolutional layer 2. The feature maps obtained from all four branches are concatenated and passed to the RNN block of the ICGNet model.

In the proposed work, we have also used filters of size 1 to pool the input (and feature maps from previous layers) across the channel dimension. Furthermore, as with other convolutional layers, a non-linearity (mostly ReLU) is used with 1 × 1 convolution, allowing it to perform significant computations on the input feature maps. As a result, 1 × 1 convolution followed by

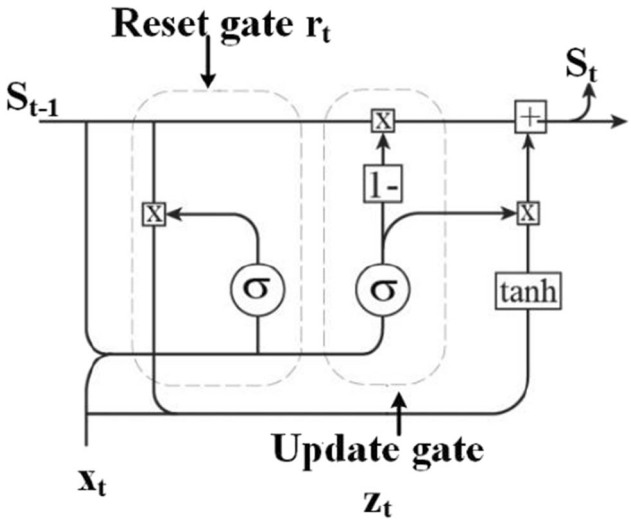


Fig. 7 GRU cell

ReLU non-linearity aids the model in learning across channels [39]. In this research, the intuition behind using 1 × 1 convolution in the 'branch 1' of the first convolution layer is mainly to learn the correlation across the features/channels. The 1 × 1 convolutions used in the second convolution layer serve a dual purpose, the first is pooling across channels, and the second is dimensionality reduction.

3.2.2 RNN block

CNNs are capable of extracting the local features and hence can capture temporal dependencies within a frame of data. But convolutional layer doesn't account for the inter-frame temporal dependencies. The time series activity data have temporal dependencies beyond the frame boundaries. To capture this temporal context contained in activity data, the use of RNNs is desirable. But the traditional RNN suffers from the vanishing gradients problem and, therefore, cannot capture long-term dependencies [7]. In the action recognition data, the long-term dependencies are important and need to be considered to precisely classify the activities. Thus, a variant of RNN called GRU is used in the proposed technique to capture the long-term dependencies. GRUs can overcome the problem of exploding and vanishing gradients [16] that existed with traditional RNN units. The RNN block of the proposed ICGNet architecture (Fig. 8) is comprised of two consecutive GRU layers. Figure 7 shows the basic GRU cell. The equations that represent the GRU cell are presented in Eqs. 1–4. The GRU cell consists of two gates, namely an update gate and a reset gate. The update gate in the GRU cell helps determine how much of the past information will be passed to the next state. This gate is updated according to Eq. (1). The reset gate is used to determine how much of the information should be discarded. It is updated according to Eq. (2). By virtue of these gates, the GRU cells can remember the significant information from the past and thus can be helpful to model the temporal context of the sequence data.

Gates:

$$z_t = \sigma(W_z x_t + U_z s_{t-1}) \tag{1}$$

$$r_t = \sigma(W_r x_t + U_r s_{t-1}) \tag{2}$$

States:

$$\widetilde{s}_t = \tanh(W_s x_t + U(r_t \odot s_{t-1})) \tag{3}$$

$$s_t = (1-z_t)s_{t-1} + z_t \widetilde{s}_t \tag{4}$$

Where, x_t is the present input, s_{t-1} is the previous output; z_t and r_t are the update and reset gates; s_t is the output from the GRU unit at timestamp 't' and $\widetilde{s}_t$ is the candidate output. W_z, W_r, W_s, U_z, and U_r are the weight matrices. s_t is updated using $\widetilde{s}_t$ and the update gate z_t decides when to update s_t. Reset gate r_t is used to calculate the candidate $\widetilde{s}_t$ new value and it tells how relevant is s_{t-1} for computing the next candidate for s_t.

The proposed model, by making use of CNN and GRU, exploits the strengths of both. CNN takes care of the extraction of local features and that too at multiple scales due to the use of multiple sized kernels, and GRU captures the long-term dependencies in the time-series data. Consequently, the proposed architecture could capture diverse information of the sensor data.

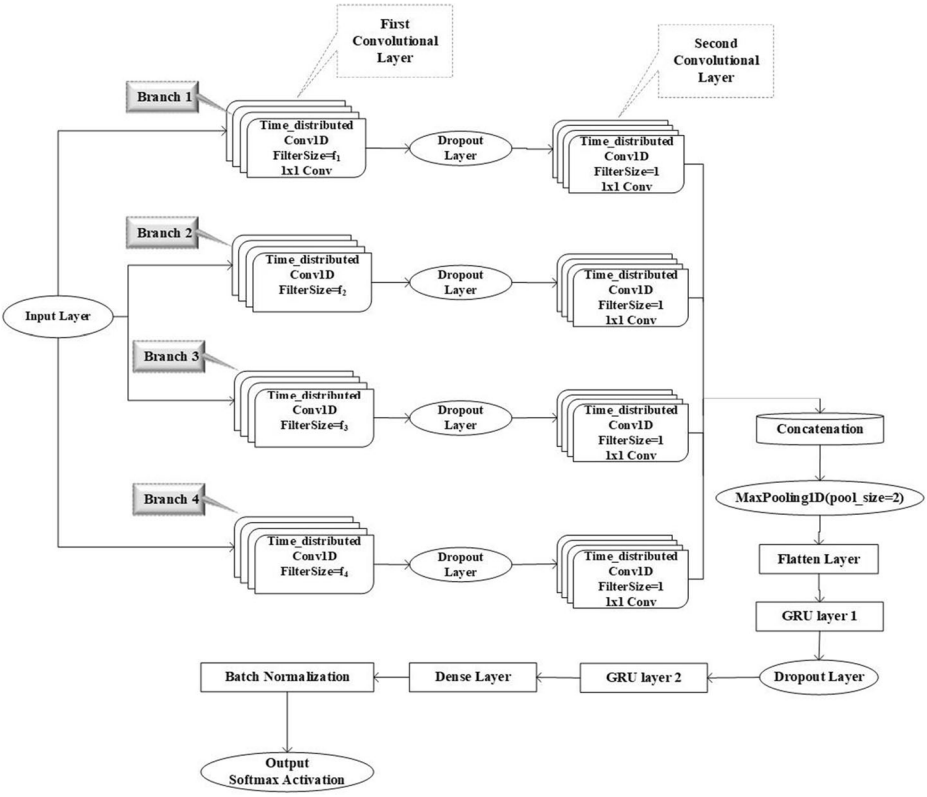


Fig. 8 Network Architecture of the proposed "ICGNet" model

3.3 Proposed ICGNet network architecture

The architecture of the proposed network is depicted in Fig. 8. The real-valued input vector, obtained after data segmentation, is passed through a 1D convolution operation. The network architecture has four parallel convolutional branches. The first 1D convolutional (Conv1D) layer of branch1, branch2, branch3, and branch4 contains 32, 64, 64, and 64 filters respectively. Each branch uses different convolutional filter sizes in the convolutional layer. Filter sizes (f_1, f_2, f_3, and f_4) of 1, 3, 5, and 11 are used in the first, second, third, and fourth branches, respectively. The use of different filter sizes simultaneously on the input data enables the convolutional layers to capture multiple local dependencies in the data. Therefore, the network can extract feature information of diverse scales. The activation function used in the convolutional layer is ReLU. The convolutional layer outputs from the first, second, third, and fourth branches are then passed through another 1D convolutional layer with a filter size of 1. The feature maps produced by the second convolutional layer from all four branches are concatenated and passed to a 1D max pooling layer with a pool size of 2. The max-pooled output is then flattened so that it can be passed to the two consecutive GRU layers. In ICGNet, the number of GRU layers is chosen to be two, as suggested in [32]. The number of units used in the first and second GRU layers is 32 and 16, respectively. The second GRU layer's output is forwarded to a dense layer with 64 units, followed by a batch normalization (BN) layer that is succeeded by a dense output layer. The output layer uses the softmax activation function,

which generates the probability distribution over all the classes of activities and classifies the input.

4 Experiments and results

The proposed network for HAR is validated using two public datasets viz. MHEALTH and PAMAP2. TensorFlow backend and Keras framework are adopted to design the proposed end-to-end classifier. The cross-entropy loss is minimized by training the model. A Keras callback option of 'ReduceLROnPlateau' is used, which monitors the validation loss parameter and reduces the learning rate (LR) by a factor of 0.2 after the 'patience' of 5 epochs. The minimum LR value is set to 0.0001. To evaluate the performance of the proposed model, it is compared with various approaches for HAR from the literature.

This section describes the evaluation metrics, details of models implemented for performance comparison, experiments performed, and results obtained. The hyperparameters employed are summed up in Table 3. All the other hyperparameters are used in this research work with their default values. All the experiments are executed on GeForce GTX 1660 Ti.

4.1 Hyper-parameter setting

The selection of hyper-parameter values is a vital part of designing deep learning models. Hyper-parameters' values are to be chosen such that the model achieves high performance. There are mainly three methods (random-based, manual, and grid-search approach) to tune the values of hyper-parameters [23]. The random-based approach uses some random set of values for these parameters. The manual approach uses the results of validation data and the experience in the field, and the grid-search approach uses a comprehensive set of values. In this paper, hyper-parameters are selected using manual-based and grid-search based techniques. To start with, a range of coarse values was selected and then based on the experience and results of the validation data, this range was further narrowed down. The hyper-parameter values obtained after this tuning are listed in Table 3.

Table 3 Hyper-parameters used for the proposed network

Phase	Hyper-parameters	Values used
Data Preprocessing	Window size (or Input vector length)	256
	Step-size	128
	Number of input channels	23 (for MHEALTH dataset) 21 (for PAMAP2 dataset)
Architecture	No. of convolution layer	2
	Filter sizes f_1, f_2, f_3, f_4	1, 3, 5, 11
	Pool_size	2
	Padding	same
	DropOut	30%
Training	Batch size	400
	Maximum number of epochs	120
	Learning rate	Initial LR 0.001 Min LR 0.0001
	Optimizer	Adam

4.2 Evaluation Metrics

The evaluation metrics used in this research work are Accuracy, F1-score, and Confusion Matrix (CM). All these metrics are defined in terms of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). TP, TN, FP, and FN [6, 18] are as defined below:

- TP: is when the sample's predicted class is the same as that of the true class of the sample.
- TN: is when the predicted class and the true class do not correspond to the searched class.
- FP: is when the sample is predicted to be of searched class when it actually belongs to a different class.
- FN: is when a sample actually belongs to a particular class but is predicted to be of a different class.

Accuracy measures the percentage of correct predictions relative to the total number of samples.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision (P) is the ratio of correctly predicted positives (TP) to the total number of samples predicted as positives.

$$P = \frac{TP}{TP + FP} \tag{6}$$

Recall (R) is the ratio of correctly predicted positives (TP) to the total number of positive samples.

$$R = \frac{TP}{TP + FN} \tag{7}$$

The F1-score measure is particularly significant in the case of HAR because human activity datasets are mostly unbalanced. F1-score is independent of class distribution; hence we evaluated the models using a weighted F1-score. It values each category's correct classification equally. F1-score is the harmonic mean of Precision and Recall values.

$$F1 - score = 2 \sum \frac{n_c}{n_t} \frac{P_c \times R_c}{P_c + R_c}. \tag{8}$$

Where, n_c is the number of samples in class c, and n_t is the total number of samples. P_c and R_c are the precision and recall values for class c.

Confusion Matrix is a table that summarizes the performance of the classifier. It is a square matrix where rows and columns respectively represent true labels and predicted labels. It gives us a complete view of how the classifier is performing and what kind of errors it is making. Thus, CM helps us to visualize the classifier's performance.

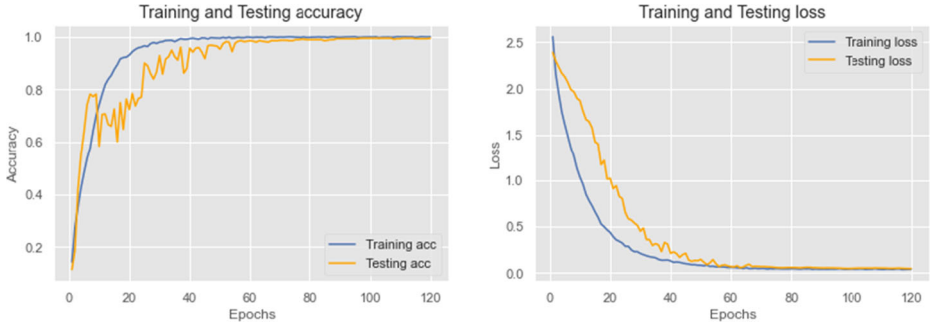
4.3 Results and discussion

The proposed ICGNet model is validated using two publicly available datasets viz. MHEALTH, and PAMAP2. The proposed model is compared with various HAR techniques proposed in literature that have reported results on MHEALTH and/or PAMAP2 datasets. Additionally, we implemented some of the HAR related state-of-the-art (SoTA) works like CNN [63], CNN-LSTM [47], stacked LSTM [44], CNN-GRU [20], and iSPLInception [53] to thoroughly compare the proposed model's performance. The same set of training and test data are used for all these models and the ICGNet model for consistency and meaningful assessment. This section presents the results of the experiments performed using both datasets and the comparisons made.

4.3.1 Results of MHEALTH dataset

The MHEALTH dataset samples are divided on the basis of the user-id. Using the window-size of 256, the total number of samples obtained is 2678, out of which 2148 samples are used for training, and 530 samples are used for testing the model. The accuracy and loss plots for training and testing obtained using the proposed ICGNet on the MHEALTH dataset are depicted in Fig. 9. The obtained CM on test set of the MHEALTH dataset is shown in Fig. 11a. From the CM, the proposed method is evident to perform well in detecting all the twelve activities, be it a simple activity (sitting, standing, etc.) or a complex activity (like cycling, knee bending, etc.).

The proposed ICGNet is compared with various SoTA HAR techniques using smartphone and wearable sensors is presented in Table 4. The performance comparison is made using the standard evaluation metrics viz. accuracy and/or F1-score. As can be seen from Table 4, the proposed model significantly outperforms the compared HAR approaches. Jalal et al. in [30] used inertial sensors data and preprocessed it using Savitzky–Golay, median and hampel filters. Several features, including binary, wavelet, and statistical features, were extracted. The MEMM was used for the highest entropy. Their technique achieved 90.91% accuracy on the MHEALTH dataset. The recall values obtained for cycling, crouching, and frontal elevation of arms were less than 90%. Moreover, their technique involved manual feature engineering and a good amount of data preprocessing. The HAR system proposed in [31] used various features viz. GMM, ECG, the MFCC, and statistical features, and used a BGWO decision tree classifier. The technique achieved an accuracy of 93.95% on the MHEALTH dataset.



a) Accuracy plot b) Loss plot

Fig. 9 Accuracy and loss plots for the MHEALTH dataset

Table 4 Performance comparison of various HAR techniques using the MHEALTH Dataset

Approaches	Accuracy (%)	F1-score (%)
Gaussian-Kernel PCA+CNN [61]	93.90	–
DT classifier + BGWO [31]	93.95	–
Ensemble algorithms [45]	94.72	94.12
Adam + Maximum entropy markov model [30]	90.91	–
CNN-pff [25]	91.94	–
LSTM-CNN [41]	95.56	–
Recurrent Convolutional Attention model [11]	94.05	–
ICGNet (Proposed model)	**99.25**	**99.28**

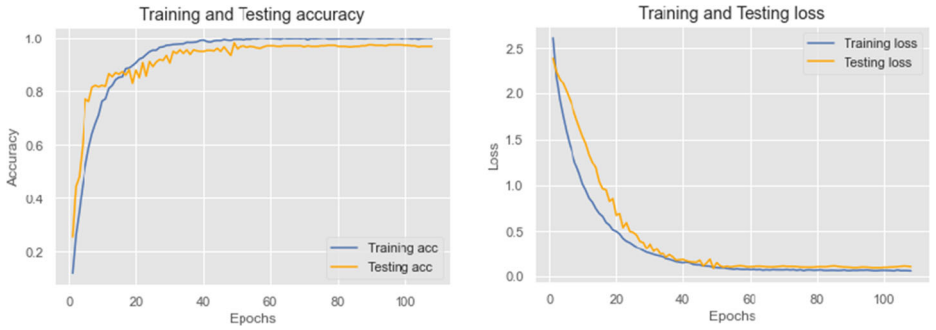
However, the method involves manual intervention to extract features, and it performed poorly in detecting activities like knee bending and frontal elevation of arms. Nguyen et al. in [45] used an ensemble of several ML techniques viz. SVM, Random Forest, MLP, LR, Naive Bayes, and KNN to boost the HAR performance. Their method achieved accuracy and F1-score of 94.72% and 94.12%, respectively. The deep CNN model 'CNN-pff' using a weight sharing mechanism proposed in [25] achieved an accuracy of 91.94%, which is 7.49% less than that of ICGNet. Also, the total number of parameters used was more than 9,00,000, which is relatively high compared to the number of parameters 235,692 used in ICGNet. Using Gaussian kernel PCA for feature extraction and a deep CNN to further use these features to classify the activities, Ha and Choi [67] achieved an accuracy of 93.90%. Still, the accuracy achieved is 5.53% less as compared to ICGNet. The LSTM-CNN model proposed by Lingjuan et al. [41] uses a hybrid of LSTM-CNN where a CNN follows an LSTM layer. Their method outperformed the baseline LSTM and CNN models in terms of accuracy value. However, the ICGNet model provides a 3.9% relative improvement over the LSTM-CNN model. Chen et al. [11] could achieve an accuracy of 94.05% on the MHEALTH dataset with their semi-supervised CNN-LSTM model. The comparison results from Table 4 indicate that the ICGNet outperforms the compared HAR approaches from the literature.

4.3.2 Results of PAMAP2 dataset

The PAMAP2 dataset samples are divided on the basis of the user-id. Using the window size of 256, the total number of samples obtained is 13,518, out of which 11,656 samples are used for training, and 1862 samples are used for testing the model. Figure 10 depicts the accuracy and loss plots for training and testing obtained using the PAMAP2 dataset.

The CM obtained on the test set of the PAMAP2 dataset is shown in Fig. 12a. From the diagonal elements of CM, it can be seen that the recall value for activity' ascending stairs' is 82%. The activity' ascending stairs' is mostly confused with 'walking.' Except for 'ascending stairs,' the model is seen to perform very well to recognize all the other eleven activities.

The ICGNet model was compared with various techniques for HAR from the literature, using the PAMAP2 dataset. Performance comparison of ICGNet and other HAR techniques from literature is made using standard performance measures of F1-score or accuracy or both, and the results of the comparison are displayed in Table 5. Chen et al. [12] used specific convolutional subnetworks to extract features from different sensors signal. Their approach, however, performed poorly in the detection of complex human activities and achieved an F1-score of 83.6%, which is quite low when compared to ICGNet's F1-score of 97.62%. Hammerla et al. [26] proposed DNN, CNN, and RNN models for HAR. They performed



a) Accuracy plot b) Loss plot

Fig. 10 Accuracy and loss plots for the PAMAP2 dataset

around 4000 experiments and established benchmark results on three datasets. The DNN, CNN, LSTM, and Bi-LSTM network they proposed, achieved an F1-score of 90.4%, 93.7%, 88.2%, and 86.8% on the PAMAP2 dataset. The conditionally parameterized convolution approach in [14] achieved test accuracy of 94.01% on PAMAP2; however, the dataset distribution between training and test set used was not based on user-id; instead, 70% of data of each class were randomly selected for the training set and the rest for the test set. The deep CNN model presented in [63] achieved an accuracy of 91%, which is 6.6% less as compared to that obtained using ICGNet. Zeng et al. [69] proposed an attention-based LSTM model. Their approach helps visualize which part of the sensor signals are being attended by the model, improving its interpretability. However, their model could achieve an accuracy value below 90%. The recurrent attention-based model introduced in [11] gives more insight into the input data's salient parts and makes the model understandable. The model designed was robust, but the accuracy achieved was 83.4% which is quite low when compared to that achieved with ICGNet. The 'Dfternet' proposed in [28] is a CNN-based approach that uses a dynamic fusion strategy that enables the model to perform well. It also used a quantization mechanism that helped it achieve desirable performance at low memory and computation requirements. However, the F1-score achieved is 6.2% lesser than that of our ICGNet model. The results presented in Table 5 show that our proposed approach outperforms the state-of-art HAR techniques using the PAMAP2 dataset by a comfortable margin.

Table 5 Performance comparison of various HAR techniques using the PAMAP2 dataset

Models	Accuracy (%)	F1-Score (%)
DEBONAIR [12]	–	83.6
DNN [26]	–	90.4
CNN [26]	–	93.7
LSTM-S [26]	–	88.2
Cond Conv [14]	–	94.01
CNN [63]	91.00	91.16
LSTM + Attention [69]	–	89.96
Recurrent Convolutional Attention model [11]	83.42	–
CNN (Dfternet) [28]	–	91.4
ICGNet (Proposed model)	**97.64**	**97.62**

4.3.3 Qualitative analysis of the ICGNet

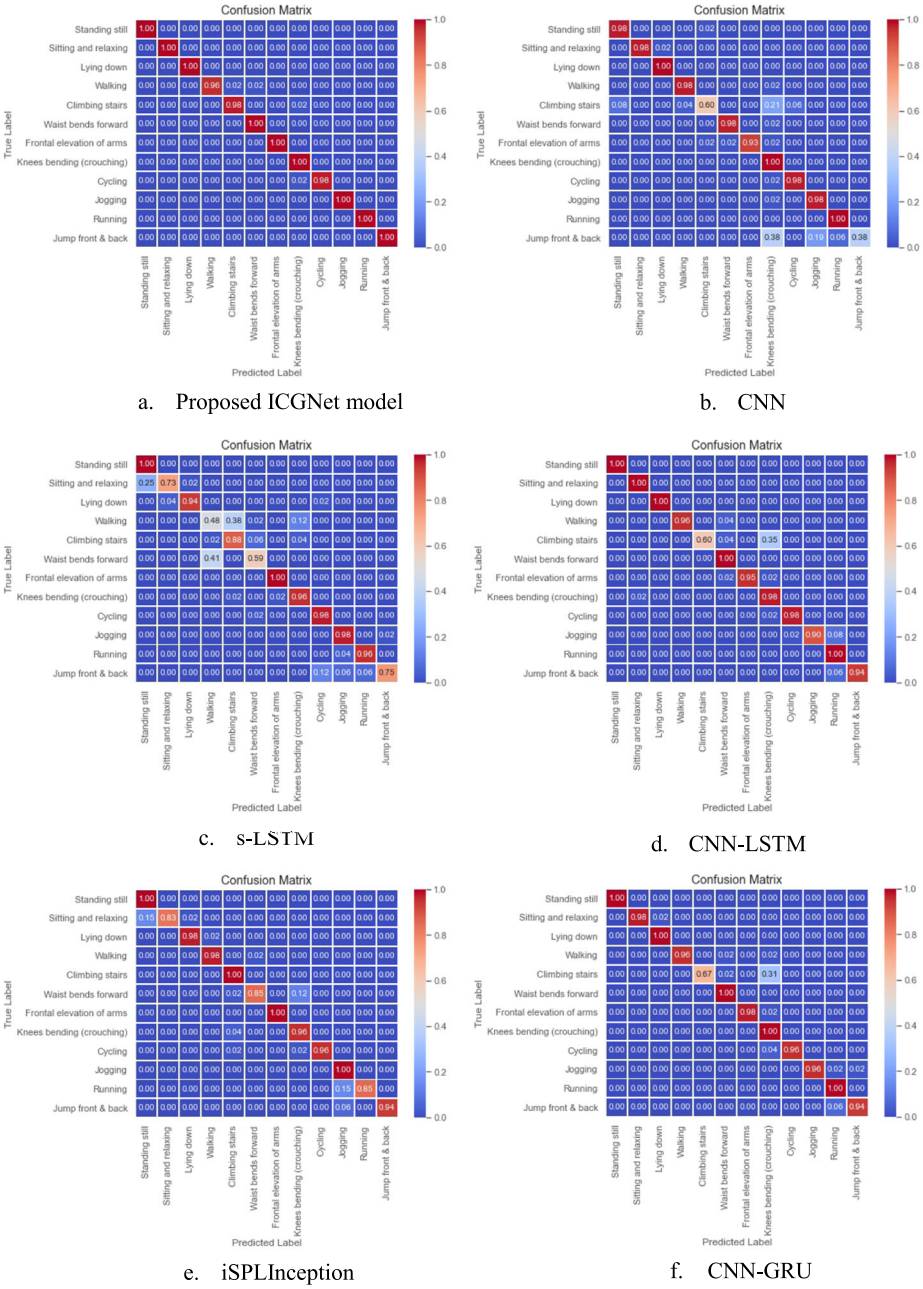
The training performance of the proposed approach on MHEALTH and PAMAP2 datasets has been represented in Figs. 9 and 10, respectively. It has been observed that the graph (training and testing) is less fluctuated, which simply indicates the learning stability during significant feature extraction. Additionally, the absence of a large gap between training and testing graphs ensures that overfitting is reduced. Further, the model has attained its highest accuracy within 50 epochs for PAMAP2 and 80 epochs for MHEALTH and remains stable throughout its learning. Similarly, the loss plot started from 2.5 and went down to approximately 0.05. This qualitative analysis validates the performance of ICGNet in terms of accuracy and loss plots.

4.3.4 Results of comparison with benchmarking models

We implemented some of the HAR related SoTA models viz. CNN [63], CNN-LSTM [47], stacked LSTM [44], CNN-GRU [20], and iSPLInception [53] to thoroughly compare the proposed model's performance. We implemented these models as per the details shared in the respective papers. The training and testing dataset used is the same as used for the ICGNet model for consistency and meaningful assessment. The performance comparison is made based on the standard evaluation metrics commonly used to gauge the performance of a classifier viz. Accuracy (A), Precision (P), Recall (R), and F1-score. The confusion matrix is also provided to get an insight into how the classifier is performing in recognition of each activity. The diagonal elements of the CM reflect the Recall value obtained for the respective activity. The total number of parameters (#param) required for each model is also used to compare the model's complexity, as the more the number of parameters required for a model, the more resources (like memory, computational requirement, training and inference time, etc.) it will require. A more resource-hungry model is not suitable to be used in real-time embedded environments.

Confusion matrices for all the methods are shared in Figs. 11 and 12 for MHEALTH and PAMAP2 datasets. Table 6 shows the values of performance metrics obtained for all the benchmark models and the ICGNet. The s-LSTM model has achieved the lowest accuracy and F1-score values among all the compared approaches. The deep convolutional LSTM approach proposed in [47] attained decent accuracy and F1-score values for both datasets. However, the number of parameters required is the second-highest when compared to other approaches.

The CNN-GRU technique in [20] also used a combination of CNN and GRU layers. Despite having similarities with it, our ICGNet sufficiently differs from it when the structure and arrangement of CNN layers are compared. Our CNN block not only uses multiple filter sizes at the same convolution level but also logically makes use of the 1×1 convolution operation to reduce the total number of parameters utilized. The CNN-GRU model shows good performance on both datasets, but compared to ICGNet, the accuracy and F1-score values attained by ICGNet comfortably surpass those achieved by the CNN-GRU model. Also, the number of parameters required by it is higher as compared to that of ICGNet. The iSPLInception model attained good accuracy and F1-score values of 94.72% and 94.79% for the MHEALTH dataset. But the total number of parameters required for iSPLInception is the highest compared to other models. iSPLInception model took the highest number of epochs (350 epochs) and training time to converge. As can be seen from the results, the ICGNet model has achieved the highest accuracy and F1-score values of 99.25% and 99.28%, respectively, for the MHEALTH dataset and 97.64% and 97.28% for the PAMAP2 dataset. Moreover, the number of parameters required for the ICGNet is comfortably less than the other compared benchmark approaches.



a. Proposed ICGNet model

b. CNN

c. s-LSTM

d. CNN-LSTM

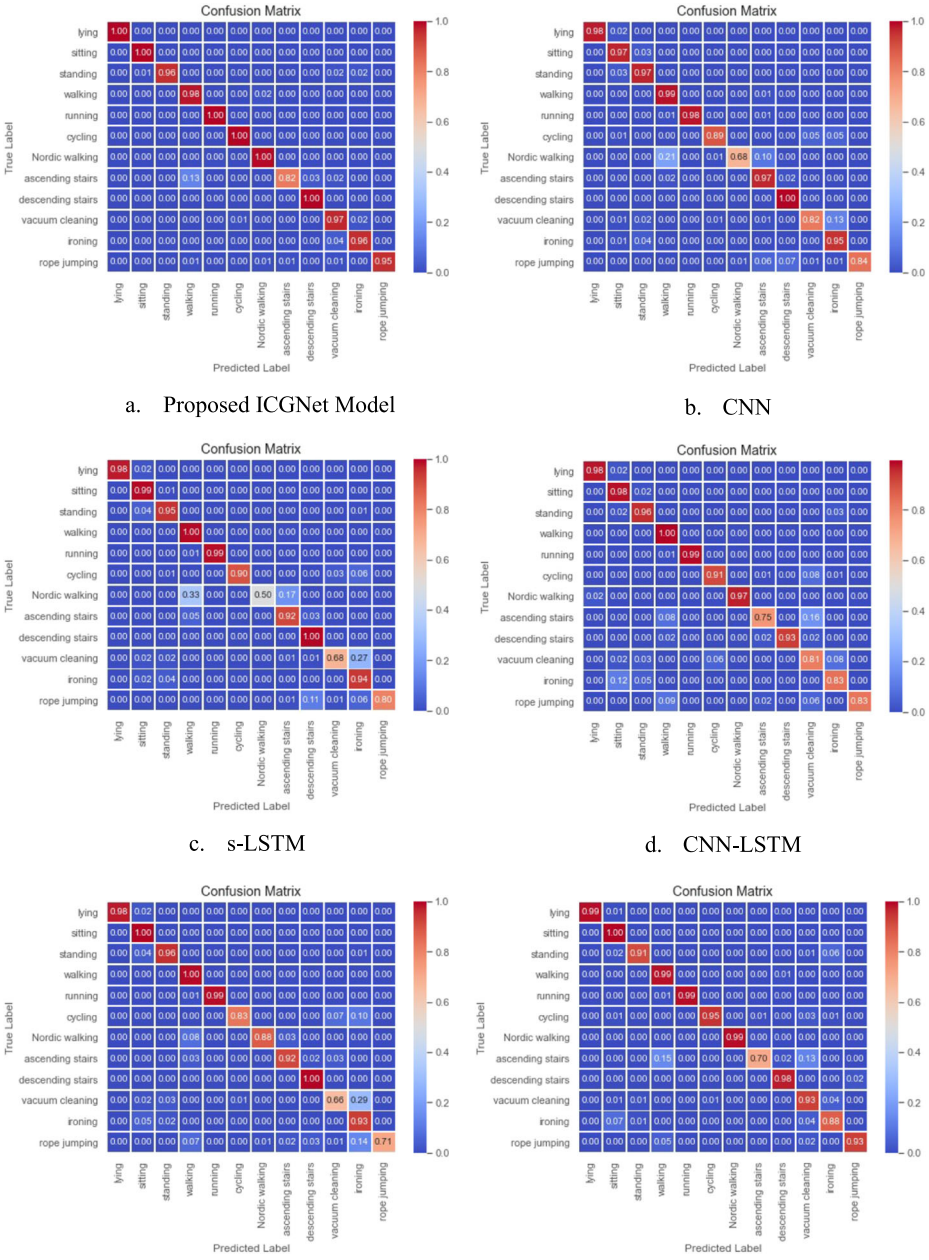
e. iSPLInception

f. CNN-GRU

Fig. 11 Confusion Matrices for MHEALTH dataset

4.3.5 Results of statistical tests

To verify the significance of performance improvement achieved with the proposed ICGNet model, the recall measure of the benchmark models and the ICGNet are compared by



a. Proposed ICGNet Model

b. CNN

c. s-LSTM

d. CNN-LSTM

e. iSPLInception

f. CNN-GRU

Fig. 12 Confusion Matrices for PAMAP2 dataset

conducting a statistical test called 'Wilcoxon signed-rank' test [21] at a significance level of 0.05. The ICGNet model is compared with the benchmark models viz. CNN [63], CNN-LSTM [47], stacked LSTM [44], CNN-GRU [20], and iSPLInception [53]. The diagonal values of the confusion matrix give the Recall values for each activity. These Recall values

Table 6 Performance comparison of the proposed model with the benchmarking DL models

Model	PAMAP2 dataset					MHealth dataset				
	P	R	Acc	F1-score	#param	P	R	Acc	F1-score	#param
CNN [63]	91.06	91.95	91.08	90.75	329,612	93.99	92.83	92.83	92.37	330,508
s-LSTM [44]	88.78	88.73	87.54	87.11	263,072	86.34	86.04	86.04	85.60	264,104
CNN-LSTM [47]	93.32	91.20	92.21	92.01	1,099,020	95.45	94.23	94.15	94.15	1,100,172
iSPLInception [53]	90.55	90.15	89.90	89.71	1,333,568	95.51	94.59	94.72	94.79	1,334,280
CNN-GRU [20]	94.85	94.63	94.63	94.59	346,728	96.11	95.30	95.28	95.19	349,416
Proposed	**97.78**	**96.93**	**97.64**	**97.28**	**235,692**	**99.27**	**99.31**	**99.25**	**99.28**	**237,356**

obtained for each model are then used to perform the statistical tests. Table 7 shows the test results obtained. The p-values obtained for ICGNet versus other models are less than 0.05, thus proving the statistical significance of the ICGNet.

4.3.6 Additional experiments for hyper-parameter tuning

The selection of hyper-parameter values is a vital part of designing deep learning models. Several experiments were performed using the MHEALTH dataset to tune the values of batch size, initial LR, number of convolution layers, first convolutional layer's filter sizes (f_1, f_2, f_3, and f_4 to be used in branches 1, 2, 3, and 4, respectively), and the window size to be used for the segmentation of time-series data obtained from the sensors.

Selection of learning rate (LR) Selection of learning rate is critical to the training process. If LR is too small, the model learns very slowly because it makes tiny updates to network parameters. A too high learning rate value will cause unnecessary divergent behavior in the loss function. Figure 13 shows the results of experiments performed using different initial learning rate values. The results show that the initial LR of 0.001 gives the optimum results.

Selection of batch size (BS) Batch size is the batch sample size after which the network's weights get updated during the training process. When the total training data is very less the batch size chosen is the same as the training data size. For large datasets, processing in batches is adopted. Increasing the BS within an appropriate range can more accurately determine the direction of gradient descent as well as lessen the training shock. However, increasing its value beyond a certain range will slow down the updating of parameters. Several experiments were

Table 7 Comparison of models based on Wilcoxon signed-rank test

Models	Wilcoxon signed-rank test (p value) MHEALTH dataset	Wilcoxon signed-rank test (p value) PAMAP2 dataset
ICGNet vs CNN [63]	0.0234	0.04
ICGNet vs s-LSTM [44]	0.002	0.041
ICGNet vs CNN-LSTM [47]	0.0312	0.0044
ICGNet vs CNN-GRU [20]	0.0156	0.0063
ICGNet vs iSPLInception [53]	0.0234	0.0293

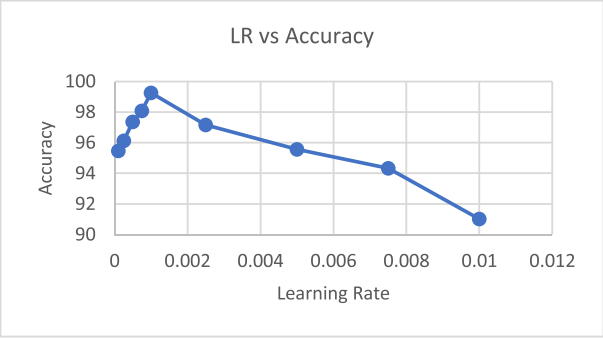


Fig. 13 Accuracy values (in %), obtained for different values of Initial LR using the MHEALTH dataset

performed by varying the batch size between 64 and 600. The results of experiments performed on different batch sizes are summarized in Fig. 14, and the results indicate that the batch size of 400 gives the highest values of accuracy.

Selection of the number of convolution layers Selection of the number of convolution layers is critical for the network's performance. Convolution layers are used to extract relevant features from the data. Multiple layers are stacked together to extract hierarchical abstractions. However, increasing the number of layers beyond a certain point causes the saturation in performance, and gradual degradation starts. Adding more layers to an appropriately deep model may cause training error to increase. To investigate the impact of the number of convolution layers, experiments were performed varying the number of Convolutional layers in the ICGNet model. Three configurations of ICGNet were tested viz.: ICGNet with two convolution layers (Fig. 6b), ICGNet with three convolution layers (Fig. 15a), and ICGNet with four convolution layers (Fig. 15b).

The number of convolution layers to be used in the ICGNet model is decided based on the performance metrics values, the total training time, and the total no. of parameters required for each model configuration. Table 8 shows that as the number of convolution layers increases, so do the number of parameters and training time, resulting in an increase in computation cost. However, increasing the number of convolution layers doesn't cause any increase in the accuracy values. The 2-layer convolution configuration of the ICGNet model is found to be optimum and hence chosen in this work.

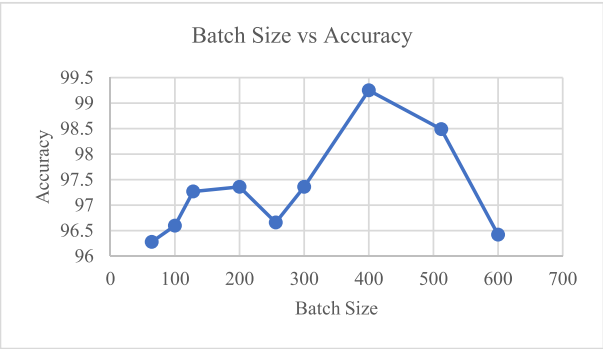


Fig. 14 Accuracy values (in %), obtained for different values of Batch Size (BS) using the MHEALTH dataset

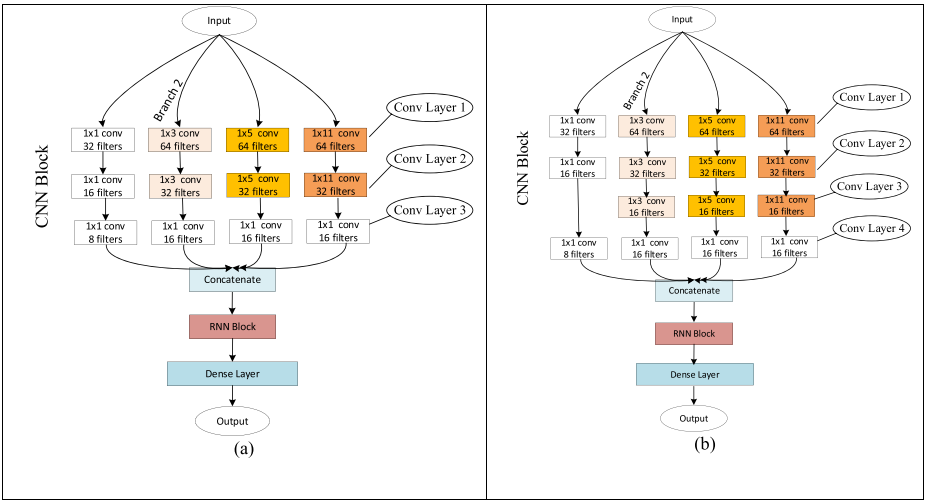


Fig. 15 **a** 3-layer convolution block in ICGNet **b** 4-layer convolution block in ICGNet

Selection of filter size Selection of filter size parameter is quite challenging because the increased size of filter size (13 × 13, 15 × 15, 17 × 17, and so on) helps to obtain feature pool in a stipulated time but fails to capture relevant features. At the same time, a smaller filter size (1 × 1) gets the high-level details of the target data but suffers from computational time. Hence choosing the correct set of filter sizes is a complex task. We selected a set of four values for window size, and for each window size we experimented with different set of filter sizes. Table 9 shows the accuracy and F1-score values obtained for varying window sizes along with different filter sizes (f_1, f_2, f_3, and f_4) and their combinations used. The f1, f2, and f3 values are fixed to 1, 3, and 5, respectively, and the value of f_4 is varied. The results in Table 9 show that the highest values for accuracy and F1-score are obtained for the window size of 256 and the filter size values of 1,3,5, and 11 for f_1, f_2, f_3, and f_4, respectively. Whereas the second highest values of accuracy and F1-score are obtained for the window size of 128 and filter sizes of 1, 3, 5, 11 corresponding to f_1, f_2, f_3, and f_4, respectively. Smaller window size of 64 didn't perform well in comparison to higher window sizes. The data, when segmented using a window size of 512, the model is observed to take almost around 250 epochs to converge. While with window sizes 128 and 256, the model takes a maximum of 120 epochs or less to converge.

The hyperparameter values thus selected through the extensive set of experiments helped the model reach its optimal performance. From the results of all the experiments, it can be said that the model proposed for HAR in this research work is able to outperform various state-of-the-art techniques for HAR using wearable and smartphones sensor data. The model is successfully validated on two benchmark datasets.

Table 8 Performance comparison of different ICGNet model configurations based on the number of convolutional layers

No. of Conv Layers	Total no. of parameters	Total training time (in seconds)	Accuracy (%)	F1-score (%)
2	**237,356**	**30.586**	**99.25**	**99.28**
3	250,452	41.217	99.06	98.92
4	259,460	46.253	98.87	98.87

Table 9 Accuracy (A) and F1-score(F1) values (in %) obtained for different sliding window size (WS) and various filter sizes combination used for f_1, f_2, f_3, and f_4 using the MHEALTH dataset

Filter sizes WS	1,3,5		1,3,5,7		1,3,5,9		1,3,5,11		1,3,5,13	
	A	F1	A	F1	A	F1	A	F1	A	F1
64	87.58	87.26	92.52	92.32	92.28	92.24	93.27	93.22	89.60	88.72
128	97.65	97.65	95.76	95.76	98.17	98.18	98.40	98.41	93.50	93.05
256	97.36	97.37	96.98	97.00	98.30	98.32	**99.25**	**99.28**	96.60	96.64
512	95.45	95.46	96.21	96.19	97.35	97.30	96.89	96.86	94.59	94.44

5 Conclusion

This work aimed to design an end-to-end classifier that performs everything from extraction of features to classify activities. Our primary focus was to develop a HAR model that is reasonably accurate and less complex so that it can be later deployed in embedded devices. The proposed ICGNet exploits the strengths of both the convolutional and recurrent neural networks and hence can capture the local correlations and long-term dependencies in the raw sensor data acquired via sensors like accelerometers, gyroscopes, etc. The ICGNet's CNN module uses multiple sized filters applied simultaneously over the input, which helps the CNN module compute more abstract features for local patches of data. The CNN module also exploits the 1×1 convolution operation to pool the input (and previous layer feature maps) across channels/features and reduce dimensionality. The network's convolutional layers are followed by GRU layers which can capture long-term dependencies of the sequence data. Hence using all these key features empowers the proposed network to capture multi-scale and diverse information in the sensor data, thus enabling it to classify the activities accurately. It performs automatic feature extraction on the raw data without using any hand-engineered features. The proposed ICGNet contains a lesser number of parameters when compared to other SoTA architecture; thus, it is computationally less expensive. The ICGNet is validated using two public datasets, and the results of experiments demonstrate that the network outperformed SoTA architectures proposed for HAR in the literature.

The proposed method deals with the individual's physical activities and doesn't address the interaction between individuals and objects. Hence, we intend to include and train the model with more complex activities to contain interactions among people and surroundings. Our future work will focus on the real-time implementation of the HAR system for fall detection and eldercare using IoT-enabled, low-cost inertial sensor-based device, which mainly focuses on elders' activities and for people with conditions like Parkinson's disease and dementia, etc.

Funding This project is funded under the schema of Early Career Award (DST No: ECR/2018/000203) by SERB, DST, Government of India.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ahad MAR, Antar AD, Ahmed M (2021) Basic structure for human activity recognition systems: preprocessing and segmentation. In: IoT sensor-based activity recognition. Springer, Cham, pp 13–25
2. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2013, April) A public domain dataset for human activity recognition using smartphones. Esann 3:3
3. Arifoglu D, Bouchachia A (2017) Activity recognition and abnormal behaviour detection with recurrent neural networks. Procedia Comput Sci 110:86–93
4. Asteriadis S, Daras P (2017)Landmark-based multimodal human action recognition. Multimed Tools Appl 76:4505–4521. https://doi.org/10.1007/s11042-016-3945-6
5. Banos O, Garcia R, Holgado JA, Damas M, Pomares H, Rojas I, Saez A, Villalonga C (December 2-5, 2014) mHealthDroid: a novel framework for agile development of mobile health applications. Proceedings of the 6th International Work-conference on Ambient Assisted Living an Active Ageing (IWAAL 2014), Belfast, Northern Ireland
6. Beddiar DR, Nini B, Sabokrou M, Hadid A (2020)Vision-based human activity recognition: a survey. Multimed Tools Appl 79:30509–30555. https://doi.org/10.1007/s11042-020-09004-3
7. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166
8. Catal C, Tufekci S, Pirmit E, Kocabag G (2015) On the use of ensemble of classifiers for accelerometer-based activity recognition. Appl Soft Comput 37:1018–1022
9. Chen YH, Hong WC, Shen W, Huang NN (2016) Electric load forecasting based on a least squares support vector machine with fuzzy time series and global harmony search algorithm. Energies 9(2):70
10. Chen Y, Zhong K, Zhang J, Sun Q, Zhao X (2016, January) Lstm networks for mobile human activity recognition. In: 2016 International conference on artificial intelligence: technologies and applications. Atlantis Press
11. Chen K, Yao L, Zhang D, Wang X, Chang X, Nie F (2019) A semisupervised recurrent convolutional attention model for human activity recognition. IEEE Trans Neural Netw Learn Syst 31(5):1747–1756
12. Chen L, Liu X, Peng L, Wu M (2020) Deep learning based multimodal complex human activity recognition using wearable devices. Appl Intell, pp.1-14 51:4029–4042
13. Chen K, Zhang D, Yao L, Guo B, Yu Z, Liu Y (2021) Deep learning for sensor-based human activity recognition: overview, challenges, and opportunities. ACM Comput Surv (CSUR) 54(4):1–40
14. Cheng X, Zhang L, Tang Y, Liu Y, Wu H, He J (2020)Real-time human activity recognition using conditionally parametrized convolutions on Mobile and wearable devices. arXiv preprint arXiv:2006.03259
15. Cho H, Yoon SM (2018) Divide and conquer-based 1D CNN human activity recognition using test data sharpening. Sensors 18(4):1055
16. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555
17. Dewangan DK, Sahu SP (2021) PotNet: pothole detection for autonomous vehicle system using convolutional neural network. Electron Lett 57:53–56. https://doi.org/10.1049/ell2.12062
18. Dewangan DK, Sahu SP (2021) RCNet: road classification convolutional neural networks for intelligent vehicle system. Intell Serv Robot 14(2):199–214
19. Dinarević, E.C., Husić, J.B. and Baraković, S., 2019, March. Issues of human activity recognition in healthcare. In: 2019 18th international symposium INFOTEH-JAHORINA(INFOTEH) (pp. 1-6). IEEE
20. Dua N, Singh SN, Semwal VB (2021)Multi-input CNN-GRU based human activity recognition using wearable sensors. Computing, pp.1-18 103:1461–1478
21. Fan GF, Qing S, Wang H, Hong WC, Li HJ (2013) Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting. Energies 6(4): 1887–1901
22. Fawaz HI, Lucas B, Forestier G, Pelletier C, Schmidt DF, Weber J, Webb GI, Idoumghar L, Muller PA, Petitjean F (2020) Inceptiontime: finding alexnet for time series classification. Data Min Knowl Disc 34(6): 1936–1962
23. Gumaei A, Hassan MM, Alelaiwi A, Alsalman H (2019) A hybrid deep learning model for human activity recognition using multimodal body sensing data. IEEE Access 7:99152–99160. https://doi.org/10.1109/ACCESS.2019.2927134
24. Gumaei A, Al-Rakhami M, AlSalman H, Rahman SMM, Alamri A (2020) DL-HAR: deep learning-based human activity recognition framework for edge computing. CMC-Comput Mater Continua 65(2):1033–1057
25. Ha S, Choi S (2016, July). Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In: 2016 international joint conference on neural networks (IJCNN) (pp. 381-388). IEEE

26. Hammerla NY, Halloran S, Plötz T, (2016) Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880
27. Huh JH, Seo YS (2019) Understanding edge computing: engineering evolution with artificial intelligence. IEEE Access 7:164229–164245
28. Yang Z, Raymond OI, Zhang C, Wan Y, Long J (2018) DFTerNet: Towards 2-bit dynamic fusion networks for accurate human activity recognition. IEEE Access 6:56750–56764
29. Ignatov A (2018)Real-time human activity recognition from accelerometer data using convolutional neural networks. Appl Soft Comput 62:915–922
30. Jalal A, Kim K (2020) Wearable inertial sensors for daily activity analysis based on Adam optimization and the maximum entropy Markov model. Entropy 22(5):579
31. Jalal A, Batool M, Kim K (2020) Stochastic recognition of physical activity and healthcare using tri-axial inertial wearable sensors. Appl Sci 10(20):7122
32. Karpathy A, Johnson J, Li F-F(2016) Visualizing and understanding recurrent networks. In: The 4th International Conference on Learning Representations Workshop
33. Kim E, Helal S, Cook D (2009) Human activity recognition and pattern discovery. IEEE Pervasive Comput 9(1):48–53
34. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Proces Syst 25:1097–1105
35. Kwapisz JR, Weiss GM, Moore S (2011) Activity recognition using cell phone accelerometers. SIGKDD Explor 12(2):74–82
36. Lara OD, Pérez AJ, Labrador MA, Posada JD (2012) Centinela: a human activity recognition system based on acceleration and vital sign data. Pervasive Mob Comput 8(5):717–729
37. LeCun, Y. and Bengio, Y., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), p.1995.
38. Li MW, Wang YT, Geng J, Hong WC (2021) Chaos cloud quantum bat hybrid optimization algorithm. Nonlinear Dynamics 103(1):1167–1193
39. Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint arXiv:1312.4400
40. Liu CL, Hsaio WH, Tu YC (2018) Time series classification with multivariate convolutional neural network. IEEE Trans Ind Electron 66(6):4788–4797
41. Lyu L, He X, Law YW, Palaniswami M (2017)Privacy-preserving collaborative deep learning with application to human activity recognition. In: CIKM '17
42. Malazi HT, Davari M (2018) Combining emerging patterns with random forest for complex activity recognition in smart homes. Appl Intell 48(2):315–330
43. Meng Y, Rumshisky A (2018)Context-aware neural model for temporal information extraction In: Proceedings of the 56th annual meeting of the Association for Computational Linguistics (volume 1: long papers)
44. Mutegeki R, Han DS (2020, February) A CNN-LSTM approach to human activity recognition. In: 2020 international conference on artificial intelligence in information and communication (ICAIIC) (pp. 362-366). IEEE
45. Nguyen HD, Tran KP, Zeng X, Koehl L, Tartare G (2019) Wearable Sensor Data Based Human Activity Recognition using Machine Learning: A new approach. arXiv, arXiv:1905.03809
46. Nguyen V, Cai J, Chu J (2019, August) Hybrid CNN-GRU model for high efficient handwritten digit recognition. In: Proceedings of the 2nd international conference on artificial intelligence and pattern recognition (pp. 66-71)
47. Ordóñez FJ, Roggen D (2016) Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115
48. Pannu HS, Ahuja S, Dang N, Soni S, Malhi AK (2020) Deep learning based image classification for intestinal hemorrhage. Multimed Tools Appl 79:21941–21966. https://doi.org/10.1007/s11042-020-08905-7
49. Park SW, Huh JH, Kim JC (2020) BEGAN v3: avoiding mode collapse in GANs using variational inference. Electronics 9(4):688
50. Ramesh S, Sasikala S, Paramanandham N (2021) Segmentation and classification of brain tumors using modified median noise filter and deep learning approaches. Multimed Tools Appl 80:11789–11813. https://doi.org/10.1007/s11042-020-10351-4
51. Rautaray SS, Agrawal A (2012, January) Design of gesture recognition system for dynamic user interface. In: *2012 IEEE international conference on technology enhanced education (ICTEE)* (pp. 1-6). IEEE.
52. Reiss A, Stricker D (2012) Introducing a New Benchmarked Dataset for Activity Monitoring. The 16th IEEE International Symposium on Wearable Computers (ISWC)
53. Ronald M, Poulose A, Han DS (2021) iSPLInception: an inception-ResNet deep learning architecture for human activity recognition. IEEE Access 9:68985–69001

54. Ronao CA, Cho S-B (Oct. 2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244
55. Saha J, Chowdhury C, Ghosh D, Bandyopadhyay S (2020) A detailed human activity transition recognition framework for grossly labeled data from smartphone accelerometer. Multimed Tools Appl 80:9895–9916. https://doi.org/10.1007/s11042-020-10046-w
56. Sajjad M, Khan ZA, Ullah A, Hussain T, Ullah W, Lee MY, Baik SW (2020) A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. IEEE Access 8:143759–143768
57. Singh R, Kushwaha AKS, Srivastava R (2019)Multi-view recognition system for human activity based on multiple features for video surveillance system. Multimed Tools Appl 78:17165–17196. https://doi.org/10.1007/s11042-018-7108-9
58. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9)
59. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: CVPR
60. Tsai TH, Huang CC, Zhang KL (2020) Design of hand gesture recognition system for human-computer interaction. Multimed Tools Appl 79(9):5989–6007
61. Uddin MZ, Hassan MM (1 Oct.1, 2019) Activity Recognition for Cognitive Assistance Using Body Sensors Data and Deep Convolutional Neural Network. IEEE Sensors J 19(19):8413–8419. https://doi.org/10.1109/JSEN.2018.2871203
62. Ullah M, Ullah H, Khan SD, Cheikh FA (2019, October) Stacked Lstm network for human activity recognition using smartphone data. In: 2019 8th European workshop on visual information processing (EUVIP) (pp. 175-180). IEEE
63. Wan S, Qi L, Xu X, Tong C, Gu Z (2020) Deep learning models for real-time human activity recognition with smartphones. Mob Netw Appl 25(2):743–755
64. Xia K, Huang J, Wang H (2020)LSTM-CNN architecture for human activity recognition. IEEE Access 8: 56855–56866
65. Yang JB, Nguyen MN, San PP, Li XL, Krishnaswamy S (2015) Deep convolutional neural networks on multichannel time series for human activity recognition. In: Proc. IJCAI, pp. 1-7
66. Yin W, Kann K, Yu M, Schütze H (2017) Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923
67. Yu S, Qin L (2018, September) Human activity recognition with smartphone inertial sensors using bidir-lstm networks. In: 2018 3rd international conference on mechanical, control and computer engineering (icmcce) (pp. 219-224). IEEE
68. Yu J, Zhang X, Xu L, Dong J, Zhangzhong L (2021) A hybrid CNN-GRU model for predicting soil moisture in maize root zone. Agric Water Manag 245:106649
69. Zeng M, Gao H, Yu T, Mengshoel OJ, Langseth H, Lane I, Liu X (2018, October) Understanding and improving recurrent networks for human activity recognition by continuous attention. In: Proceedings of the 2018 ACM international symposium on wearable Computers (pp. 56-63)
70. Zhao Y, Yang R, Chevalier G, Xu X, Zhang Z (2018) Deep residual bidir-LSTM for human activity recognition using wearable sensors. Math Probl Eng 2018:1–13
71. Zheng Y, Liu Q, Chen E 2014 Time series classi_cation using multi-channels deep convolutional neural networks. In: Proc. Int. Conf. Web-Age Inf. Manage. Cham, Switzerland: Springer, pp. 298-310