Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07911-0

ORIGINAL ARTICLE

Human activity recognition from sensor data using spatial attention-aided CNN with genetic algorithm

Apu Sarkar[1] · S. K. Sabbir Hossain[1] · Ram Sarkar[1]

Received: 8 March 2022 / Accepted: 29 September 2022

Abstract
Capturing time and frequency relationships of time series signals offers an inherent barrier for automatic human activity recognition (HAR) from wearable sensor data. Extracting spatiotemporal context from the feature space of the sensor reading sequence is challenging for the current recurrent, convolutional, or hybrid activity recognition models. The overall classification accuracy also gets affected by large size feature maps that these models generate. To this end, in this work, we have put forth a hybrid architecture for wearable sensor data-based HAR. We initially use Continuous Wavelet Transform to encode the time series of sensor data as multi-channel images. Then, we utilize a Spatial Attention-aided Convolutional Neural Network (CNN) to extract higher-dimensional features. To find the most essential features for recognizing human activities, we develop a novel feature selection (FS) method. In order to identify the fitness of the features for the FS, we first employ three filter-based methods: Mutual Information (MI), Relief-F, and minimum redundancy maximum relevance (mRMR). The best set of features is then chosen by removing the lower-ranked features using a modified version of the Genetic Algorithm (GA). The K-Nearest Neighbors (KNN) classifier is then used to categorize human activities. We conduct comprehensive experiments on five well-known, publicly accessible HAR datasets, namely UCI-HAR, WISDM, MHEALTH, PAMAP2, and HHAR. Our model significantly outperforms the state-of-the-art models in terms of classification performance. We also observe an improvement in overall recognition accuracy with the use of GA-based FS technique with a lower number of features. The source code of the paper is publicly available here.

Keywords Human activity recognition · Continuous wavelet transform · Deep learning · Spatial attention · Genetic Algorithm · Feature selection · Filter method

1 Introduction

Human activity recognition (HAR) is an emerging topic of research in the larger fields of ambient computing and context-aware computing. Recognizing daily life activities is becoming increasingly important in pervasive computing with lots of applications like intelligent surveillance systems [1], healthcare [2], abnormal behavior detection [3], human–computer interaction [4, 5], aid to elderly people to improve the quality of their lives, etc. HAR frameworks provide a way to sense, recognize, and classify specific movements or activities of a person using the data obtained from various sensors. A typical supervised HAR framework can be divided into basic blocks consisting of sensor data accusation, dividing the raw data into fixed-size windows, feature extraction, and finally classification. Each and every activity is represented by one or more fixed-size feature vectors extracted in the feature extraction step. These feature vectors are used for training the classifier.

Based on the usages of the sensor, HAR can be mainly categorized into vision-based HAR and wearable sensor-based HAR. The vision-based technique recognizes and classifies activities by analyzing video or images [6, 7] captured using a camera. Though vision-based techniques have a mature theoretical basis, these techniques have

✉ Ram Sarkar
ram.sarkar@jadavpuruniversity.in

Apu Sarkar
apusarkar2195@gmail.com

S. K. Sabbir Hossain
deepsabbir1999@gmail.com

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

Published online: 26 October 2022

various limitations like ambient light, camera position, potential obstacles, and invasion of privacy issues, which make them difficult in real-life applications. Wearable sensors and inertial sensors of smart devices nowadays become the more promising ways of collecting human activity data as these are easy to use, small in size, and non-intrusiveness on subjects. Besides, these sensors have none/low installation cost and low energy consumption. Smartphone and smartwatches have become a convenient option for HAR as it comes with various embedded sensors like accelerometer, gyroscope, magnetometer, compass, etc.

For the activity prediction tasks, generalizing any model for different activities and sensors is a very challenging task. Based on humans, the activity signal pattern may vary significantly as different humans perform these activities differently. Even the same activity can have different signal patterns as any specific human can do the same activity differently at a different time. Similarly, different activities can have similar signal patterns, which makes the activity classification task more confusing and challenging.

In the recent past, researchers have introduced several handcrafted feature extraction methods to extract various spatiotemporal features from the raw sensor data. The traditional supervised machine learning techniques—Support Vector Machine (SVM) [8–10], K-Nearest Neighbors (KNN) [11–13], Decision Tree [14], Ensemble approach [15, 16]—are used for classification. However, there are certain limitations of this approach like the requirement of domain expertise and rigorous data pre-processing. Also, failing to establish a proper spatial and temporal relationship among handcrafted features limits the flexibility of these approaches.

Recently, deep learning techniques gain more popularity among researchers. Ability to detect various features automatically from the raw data and to learn various deeper low levels of features gives deep learning techniques an edge over the traditional machine learning techniques. Several deep learning models are successfully applied in different areas like natural language processing [17], image segmentation [18], classification [19], etc. Specifically, convolutional neural networks (CNNs) are well known for producing outstanding results in image recognition [20, 21]. However, reformulating features of time series data as visual clues have raised much attention among computer scientists [22]. The most successful way is to describe features as visual cues [23]. Time series data can be encoded into corresponding activity images using supervisory and non-hyper-visual learning techniques in computer vision to enable deep learning techniques, specifically CNNs, to perform image recognition.

It is to be noted that a feature extraction procedure may produce some irrelevant or redundant features which increase the overall feature space. This is also true for the feature vectors produced by the deep learning model. Hence, these irrelevant features must be eliminated in order to ensure a good classification accuracy and less computational time. A feature selection (FS) algorithm tries to improve the performance of a learning algorithm and decrease the time and space requirements. FS algorithms can be divided into two categories: wrapper and filter. A wrapper method uses a classifier to calculate the fitness of each candidate solution (i.e., a subset of features) and thereby select the subset of features that has the best fitness score. On the other hand, filter-based methods rank the features in order of their importance and eliminate the less important features. Since filter methods do not need a learning algorithm, they tend to perform faster than the wrapper methods in general. However, wrapper methods are known to generate better classification accuracy than filter methods [24]. FS is an NP-hard problem as there can be 2n possible solutions for a feature space containing 'n' no of features. Determining the best solution from all the possible solutions is not a feasible option as the computational time required would be quite high. Hence, an alternative and feasible solution is to perform a guided search over the entire feature space using a heuristic strategy. This will not only decrease the computational time significantly but also produce a near-optimal solution.

In this paper, we have proposed an architecture that encodes sensor data into corresponding images and a model that enables HAR to be carried out using a spatial attention-aided CNN model in image recognition. However, the feature set, produced by this CNN model, is quite large in size. To this end, we have proposed an FS approach for selecting the optimal feature subset by eliminating the irrelevant feature attributes which also saves computational time and memory. This implies that we have used the said CNN model as the deep feature extractor only. For FS, a modified version of Genetic Algorithm (GA) [25] is used. Rather than utilizing a time-consuming classifier in each iteration, we have utilized three filter techniques specifically Mutual Information (MI) (entropy based), ReliefF (distance based), and Minimum Redundancy Maximum Relevance—mRMR (statistic based). These three methods rank the features obtained from the CNN model. We have re-ranked the features using the mean of the ranks of the features given by three filter methods. These ranks are used as the fitness of the candidate solutions (i.e., chromosomes). We have also proposed a guided mutation strategy which aims to increase the fitness of the individual chromosomes. The reduced feature set is then fed to the KNN classifier for predicting the accuracy of the overall HAR model.

The key **contributions** of the proposed work are as follows:

1. We have proposed a unique image encoding framework based on Continuous Wavelet Transform (CWT) to represent the sensor data into the corresponding spatiotemporal representation.
2. A spatial attention-aided CNN model is used to extract image features from the encoded images.
3. In order to reduce computational overhead, we have introduced a modified GA-based feature selection framework that uses three filter-based methods to determine the fitness of each candidate solution.
4. We have also proposed a guided mutation technique as an improvement over random mutation to increase the fitness score of each candidate solution.

The rest of the paper is structured as follows. Section 2 describes some relevant methods proposed by the other researchers. Details of the proposed method are mentioned in Sect. 3. In Sect. 4, we have reported the results of the proposed model while evaluated on five benchmark HAR datasets. In Sect. 5, we further discuss our findings. Finally, we have concluded the paper in Sect. 6.

2 Related work

Deep learning-based models have achieved outstanding results in a variety of fields including HAR as mentioned in recent surveys [26, 27]. Many state-of-the-art models have been developed using various deep learning techniques like CNN, Recurrent Neural Network (RNN), etc.

CNN models showed lots of promise and achieved higher recognition accuracy than other state-of-the-art methods. Nair et al. [28] used the Temporal CNN architecture, a class of temporal models that used a hierarchy of temporal convolutions, which was able to take variable-length sequence data and learn long-term dependencies. Münzner et al. [29] proposed a CNN-based sensor fusion technique to solve the problems of normalization and fusion of multimodal sensors. In [30–34], authors have used various CNN architectures to improve the recognition accuracy of HAR. Ensemble of CNN models is found in [35–37] which aims to achieve better performance than the individual models.

RNN, another deep learning technique, was also extensively used by many researchers for HAR. RNN has the special ability to learn sequences of spatial data. Like, long short-term memory (LSTM)-based networks can learn long-term dependencies from any sequences of data which make it more applicable in wearable/inertial sensor-based HAR. Preeti Agarwal and Mansaf Alam [38] developed a lightweight model using shallow RNN combined with LSTM for activity recognition. Authors in [39–42] used LSTM-based architectures to learn spatiotemporal features for the classification of human activities. Researchers also proposed various hybrid models like the combination of CNN-RNN [43], CNN-LSTM [44–48], LSTM-CNN [49], CNN-GRU (Gated Recurrent Unit) [50], and achieved significant improvement in recognition accuracy.

Inspired by the recent success of deep learning techniques especially CNN in computer vision, encoding time series data as images gain more acceptance among researchers. This method allows the machine to visually recognize and classify by learning visual patterns and structures. Zhiguang Wang and Tim Oates [22] introduced two frameworks for encoding time series data as images known as Gramian Angular Field (GAF) and Markov Transition Field (MTF). They used Tiled CNNs to classify the single GAF and MTF images as well as the compound GSF-MTF images. The authors in [51] found that varied time series features are not evident in the temporal domain but present in the frequency domain. As an alternative graphical representation for time series classification, they investigated the use of recurrence plots and proposed a method capable of extracting texture features from that graphical representation and used those features to classify time series data. In their work, Garcia-Ceja et al. [52] proposed a similar approach. They modeled the physical activity as a set of recurrence plots' distance matrices to capture temporal patterns in the signal. Afterward, a CNN was used to classify the distance matrices and obtain the final prediction. In [53], the authors experimentally found that image representation of time series data introduces different feature type that was not available in1D sensor data. Hence, they first encoded the sensor signal as a 2D texture image using a recurrence plot to visualize the recurrent nature of a trajectory through phase space. Then, they used a CNN model to learn different levels of features from the texture images. To address the variability in the distinctive region scale and sequence length, Zhang et al. [54] proposed two stages approach, where firstly they encoded the sensor data using Multi-scale Signed Recurrence Plots (MS-RP), an improvement in recurrence plot, and then applied a Fully Convolutional Networks and ResNet to handle these images. Hur et al. [55] proposed a novel encoding technique for converting an inertial sensor signal into an image with minimal distortion, namely Iss2Image (Inertial sensor signal to Image). Iss2Image divided real-valued sensor reading into three parts: integers, first two decimal places, and the next two decimal places, and then encoded as a three-channel image. Finally, a CNN model was used for image-based activity classification. Another similar encoding technique was proposed by Daniel et al. in [56]. The proposed INIM framework first encoded the sensor's signal into 3D RGB images and then used a residual network trained on the ImageNet dataset [57] for activity recognition. Qin et al. [58] introduced a novel method to encode time series data into two-channel GAF images by unifying global and local time series features. Then, they presented a fusion ResNet framework,

which learned the generated GAF image pixels correspondences between acceleration and angular velocity features. Almost similar work was done by the authors in [59]. Contrary to the previous work, they used four different types of activity images and made each one multimodal by convolving it with two spatial domain filters: the Prewitt filter and the high-boost filter. ResNet-18 was used to extract the deep features from multi-modalities and fused by canonical correlation-based fusion. Finally, a multi-class SVM was used for activity recognition. In [60], the authors have implemented the idea of transforming the 1D signal into 2D using Fast Fourier Transform (FFT). This frequency-domain image was called the spectrogram, which represents the composition of a signal from several frequencies over time and acts as an input to a three-layered CNN model for features extraction and classification. Lawal et al. [61] in their work encoded sensors signal into spectrogram using Short-Time Fourier Transformation (STFT). A simplified two-stream VGG-Net [20] like CNN architecture was proposed for activity and location recognition.

A few researchers have also tried to choose the relevant features utilizing various FS-based techniques [62, 63] for improving the overall accuracy in the field of activity recognition. Buenaventura et al. [64] proposed a HAR model based on sensor fusion in smartphones which used a filter-based method to rank the features. An enhanced HAR method was proposed by Fan et al. [65] where Bee Swarm Optimization (BSO) with a deep-Q-network was used. Dewi et al. [66] performed a comparative study on HAR datasets using four classifiers namely Random Forest (RF), SVM, KNN, and Linear Discriminant Analysis (LDA) from which it was concluded that RF has the highest accuracy. Nguyen et al. [67] proposed a position-based FS method for body sensors for daily activity recognition. Filter-based methods were used to reduce the feature set followed by a correlation-based optimization and a classifier to determine the overall accuracy of the proposed method.

GA is one of the oldest and most widely used meta-heuristic algorithms which have been explored by numerous researchers in various domains such as image contrast enhancement, class imbalance, stock price prediction, image segmentation, medical diagnostic, image steganography, feature selection, etc. Saitoh [68] proposed an image contrast enhancement technique based on GA that assessed an individual's fitness by evaluating the intensity of spatial edges included in the image. GA was used to search for a solution in global space, and the original gray image was converted to a contrast-enhanced image by observing the relationship between the input and output gray levels. In [69], an efficient image contrast enhancement using GA and fuzzy intensification operator was proposed which improved the visibility information of an image by manipulating the image intensity information. A novel oversampling approach was introduced by Arun et al. [70] to address the class imbalance problem using GA. Synthetic samples of the minority class are generated based on the distribution measure which ensures that the samples are efficient and diverse within each class. Experimental results indicated that GA-based oversampling approach improved the fault prediction performance and reduced the false alarm rate. Ha et al. [71] proposed a novel under-sampling method using GA for imbalanced data classification. The performance of the prototype classifier was maximized by minimizing the loss between distributions of original and undersampled majority objects. A novel method for stock market forecasting with Artificial Neural Network (ANN) and GA was proposed by Sharma et al. [72]. The dataset was partitioned into training, testing, and validation sets, and the stock data of COVID-19 period were used for model validation. Furthermore, in [73] a combination of GA and LSTM was proposed for stock prediction. In the initial step, GA was used to obtain ranked important factors, and finally, the optimal factors along with LSTM were used for prediction. Chun et al. [74] proposed a robust image segmentation using GA with a fuzzy measure. A fuzzy validity function was proposed which measured the degree of separation and compactness within the finely segmented regions. To maximize the quality of regions obtained by split and merge processing, a usable region segmentation was searched using GA. In [75], an image segmentation method with GA was proposed where GA was used for segmenting the images into four gray classes. A cardiovascular disease prediction using GA and neural network was proposed by Amma [76] where the weights of the neural network were determined using GA which provided a good set of weights in a few iterations. Initially, the dataset was pre-processed followed by training the system and storing the final weights which were finally used for predicting the risk of cardiovascular disease. Uyar et al. [77] proposed a GA-based trained recurrent fuzzy neural network (RFNN) method for the diagnosis of heart diseases. Hossain et al. [78] introduced a secured image steganography method based on GA and ballot transform for the integrity of important files over internet. In addition to achieving a good accuracy, various parameters such as precision, F-score, probability of misclassification error, mean square error, etc. were also calculated.

Owing to the success of GA in solving various complex optimization problems, many researchers have used GA for the FS purpose which is a binary optimization problem. Some areas where GA is used as an FS method are: microstructural image classification [79], cancerous gene identification [80], handwritten Devanagari numeral recognition [81], handwritten Bangla word recognition [82], handwritten Bangla, Devanagari and Roman numeral

classification [83], video and sensor-based HAR [62], etc. Rostami et al. [84] developed a novel community-based FS method to group similar features into feature clusters. This method predicted the number of feature clusters automatically, hence eliminating the need to determine it beforehand. GA is then applied to select the optimum subset of features by defining an objective function with an importance value attached to each feature subset. In [85], a novel cancer classification technique was proposed using deep learning and GA. It was applied to determine and classify the cancer types from the publicly available gene expression data. Tian et al. [86] proposed deep learning model selection framework based on GA for visual data classification. The process of identifying the most relevant and useful features generated by pre-trained models for different tasks was automated by the framework. In [87], a deep learning method was developed to classify different brain activities along with GA to eliminate the redundant features. Various deep learning models, namely X_axis Classification Model (XCM), Y_axis Classification Model (YCM), and Z_axis Classification Model (ZCM), were used for this purpose. These models were used to classify among the vision, movement, and forward brain activities followed by an effective combination method based on GA and Genetic Weighted Summation (GWS) rule. In 2019, Ghosh et al. [88] introduced a combination of GA and PSO for feature selection which utilized the exploitation ability of GA with the exploration capacity of PSO. Guha et al. [89] proposed a deluge-based GA to strengthen the exploitational ability and performed good on the well-known UCI datasets. In 2021, kilicarslan et al. [90] proposed a hybrid model based on GA and deep learning for nutritional anemia disease classification. GA was used to optimize the hyperparameters of Stacked Autoencoder (SAE) and CNN models. The proposed method achieved an accuracy of 98.50% when applied on real anemia dataset. Ince [91] proposed a deep learning and GA-based intelligent and automatic content visualization system. The method segmented the input image into panoptic image instances and used these to generate new images using GA. The results proved that the said method was efficient to create visually enhanced content for digital use.

Motivations: From the above discussion, it can be concluded that many researchers around the world have tried to classify human activities by analyzing the activity images. It can be observed that recognizing human activities from sensor data has always been an interesting and challenging task. Some activities such as running and walking are easy to recognize. However, there are some complex activities which are relatively difficult to classify. Developing an efficient activity recognition model can lead to the development in many potential fields such as health, sports, and understanding the psychological state of a person. For this purpose, machine learning and deep learning-based methods contributed significantly to the development of competent HAR models. However, many of these methods use heavy networks (mainly deep learning-based methods) and some even produced lower classification accuracy due the use of some irrelevant features. On the contrary, FS-based techniques not only speed up the process (i.e., take less computational time) but also increase the classification accuracy significantly. However, wrapper-based FS techniques which use a learning algorithm are slower than filter-based methods. Keeping the above facts in mind and to further speed up the process, a modified version of GA method is proposed here, which uses three filter-based methods to calculate the fitness of the chromosomes that effectively acts as the fitness function of GA. The proposed method has been evaluated on five publicly available datasets. It is observed that this method is much faster than the traditional GA, and the overall framework also outperforms many existing methods in terms of classification accuracy.

3 Proposed method

Here in this section, we first briefly discuss the proposed activity image encoding technique. Then, we explore the features extraction process from the encoded images. Finally, we present the proposed novel FS technique used for HAR. Figure 1 shows the working procedure of the proposed framework.

3.1 Continuous wavelet transform

Wavelet transform has been applied in time–frequency analysis and spatial domain signal analysis over the years, and this is one of the most effective mathematical tools used for signal processing. A wavelet transform is a signal convolution with a set of functions derived from translations and dilations of a primary function. The primary function is known to as the mother wavelet, and the translated or dilated functions are referred to as wavelets.

A wavelet is a rapidly decaying wave-like oscillation defined as function $\psi(t)\epsilon L^2(R)$ with a zero mean and exists for a finite duration, localized both time and frequency. By scaling and translating this wavelet $\psi(t)$, we can produce a family of wavelets by using Eq. (1) as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{1}$$

where $a, b\epsilon R$ and $a > 0$. a is known as the scaling parameter, and b is the transitional value. The wavelet

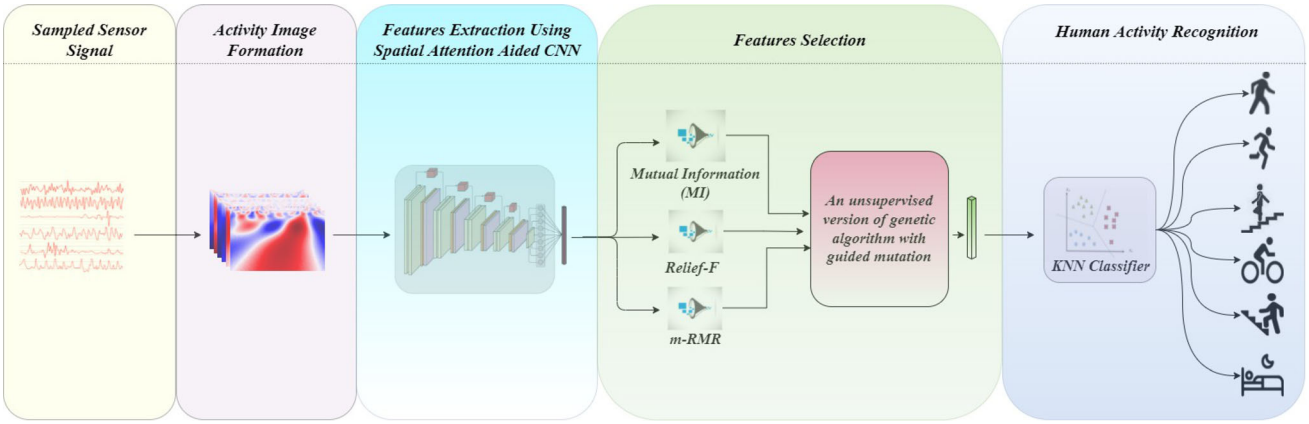

Fig. 1 Overall workflow of the proposed HAR framework

transform of a continuous signal with respect to wavelet function $\psi(t)$ is defined as Eq. (2)

$$W_x(a,b) = \int_{-\infty}^{+\infty} x(t)\psi^*_{a,b}(t)\mathrm{d}t \tag{2}$$

where $x(t)$ is a time-domain signal; $\psi^*_{a,b}(t)$ is the complex conjugate of mother wavelet. From Eqs. (1) and (2), we get Eq. (3), which defines the CWT as

$$X_w(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} x(t)\psi^*\left(\frac{t-b}{a}\right)\mathrm{d}t \tag{3}$$

CWT is nothing but the inner product of signal $x(t)$ with a continuous wavelet $\psi(t)$ scaled by parameter a and translated by value b. The pseudocode for the CWT is shown in Algorithm-1.

Algorithm 1: Pseudocode for Continuous Wavelet Transform (CWT)

Input: $1D\ time-series$: *list of fixed timestamps*
wavelet : *a function*
scale : *list of positive numbers*

Output: *2D coefficient matrix of size* $s \times s$, *where* s *is the length of the scale parameter*

Procedure:

1. Take the wavelet and compare it to a section at the starting of the original time-series signal.
2. Compute the inner product of the wavelet and the signal.
3. Shift the wavelet to the right and repeat steps 1-2 until the signal is processed.
4. Scale (stretch/shrink), the wavelet, and repeat steps 1–3.
5. Repeat steps 1 to 4 for all available values present in scale.

The outputs of the CWT are CWT coefficients, which reflect the similarity between the analyzed signal and the wavelet. These coefficients can be represented as a 2D image equivalent to the power spectrum, where time and scale/frequency are the 2 dimensions. However, the CWT coefficients depend on the choice of the mother wavelet.

One of the main advantages of wavelet transform is the presence of a wide variety of wavelets to choose from that best match the shape. In this work, we use the Gaussian Derivative Wavelets, specifically fifth-order derivatives of the function given in Eq. (4)

$$\psi(t) = C\exp^{-t2} \tag{4}$$

where C is the order-dependent normalization constant.

The fifth-order Gaussian Derivative wavelet is a real-valued odd function, which is anti-symmetric around zero. The shape of the fifth-order Gaussian Derivative wavelet and various scaled wavelets is shown in Fig. 2.

As the wavelet is a real-valued function, hence the imaginary part of the wavelet is zero.

3.2 Inertial sensor to image encoding using CWT

In order to encode the raw sensor time series data into an image form, we use the 1D CWT, which takes 1D time series as input and generates a 2D frequency-time domain scalogram. This scalogram is nothing but the CWT coefficients. Figure 3 depicts the image encoding process.

Performing CWT on the entire time series dataset is practically infeasible. Hence, instead, we perform CWT on each sample of size $t \times c$ where t is the number of timestamps and c is the total number of sensor channels. The pseudocode for CWT-based image encoding technique is given in Algorithm-2. The value of t and c varies from dataset to dataset. Each of the channels in c is a 1D time series and acts as the input to the CWT. We use t as the scale parameter. For each such sensor channel, we get a $t \times t$ scalogram as the output. Hence, for one sample, we get a c-dimensional $t \times t$ scalogram where each dimension corresponds to each sensor channel.

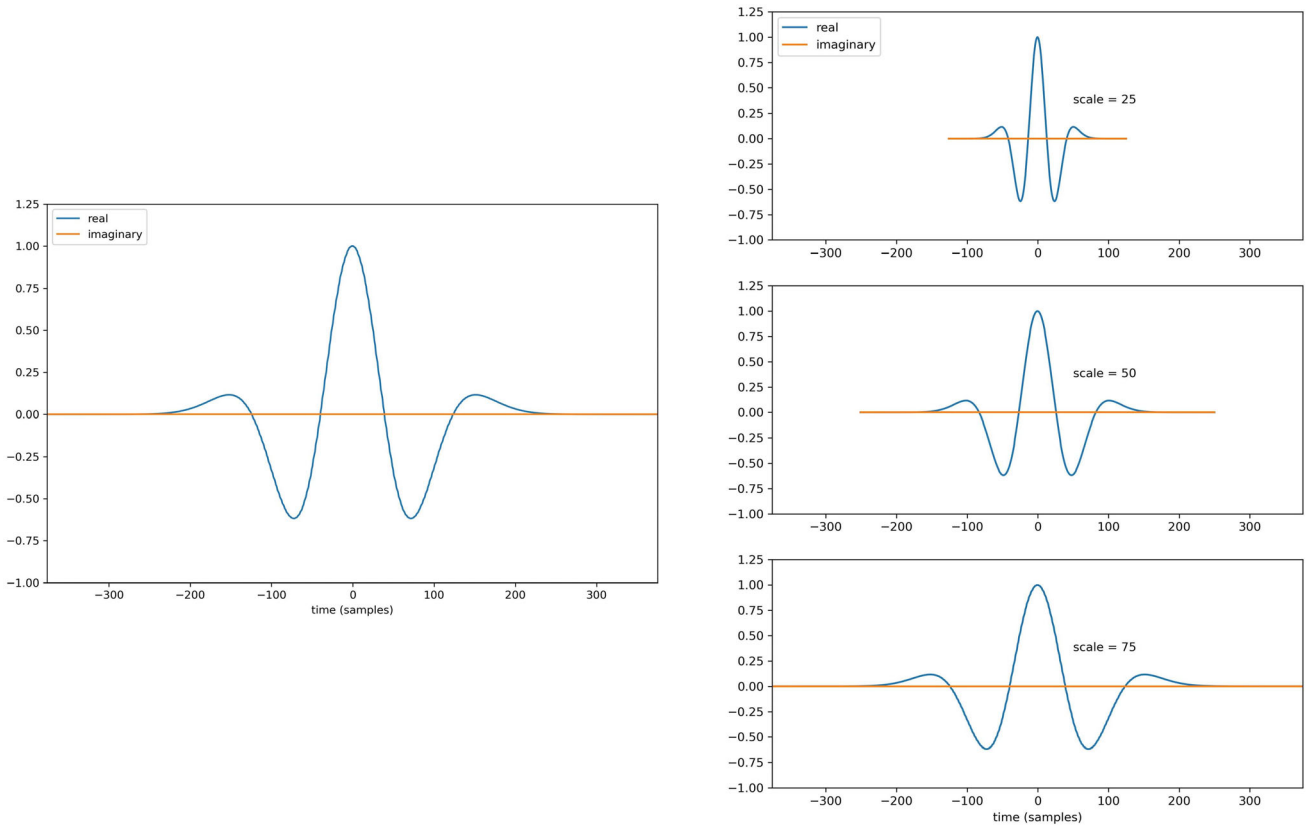

Fig. 2 Fifth-order Gaussian Derivative wavelet and its scaled version

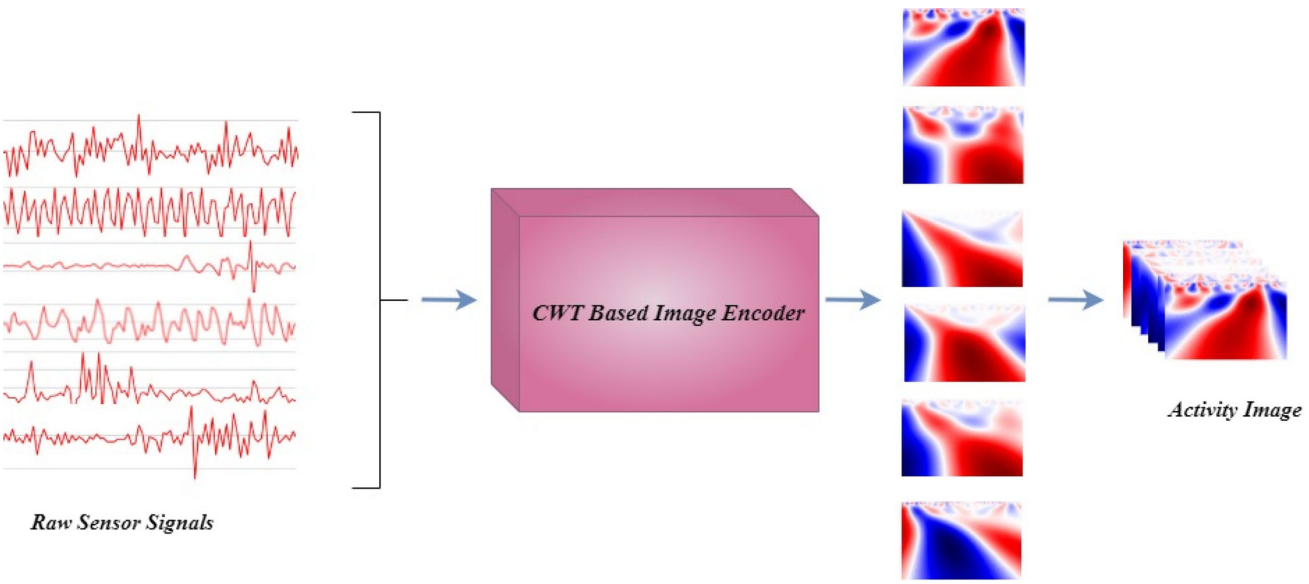

Fig. 3 Illustration of the image encoding process

Algorithm 2: Pseudo code for image encoding

Input: *A sampled dataset D, where each activity is sampled using a fixed-size overlapping window*
Output: *Images, collections of the corresponding encoded images as 4-dimensional array*
Start
$n \leftarrow$ total no. of samples/activities in the dataset D.
$s \leftarrow$[1, 2, 3, ..., t] ▷ t is no. of timestamps in each activity sample.
$c \leftarrow$ no. of sensor channels in each activity sample.
images $\leftarrow$array of shape $n \times t \times t \times c$.
wavelet $\leftarrow 5th$ order Gaussian Derivative wavelet.
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** c **do**
    signal $\leftarrow$ D[i , : , j] ▷ extracting each channel
    coeff $\leftarrow$ **CWT(signal, wavelet, s)** ▷ finding CWT coefficients
    images[i , : , : , j] $\leftarrow$ coeff[: , : t] ▷ storing the images as an array
  end
end
End

Based on the above-mentioned way, we encode each and every activity of a dataset as a $t \times t \times c$-dimensional image.

3.3 Features extraction using spatial attention-aided CNN

A CNN is large a deep neural network that simulates and understands stimuli as the visual cortex of the brain processes. A typical CNN model can be thought of as a combination of two components: the features extractor part and the classification part. The hidden layers are the CNN's features extractor, which consists of a series of convolution layers followed by pooling layers that try to detect complex features and patterns belonging to the image of a particular class by convolving with various filters followed by sub-sampling. The classification part then utilizes these features and computes the prediction probabilities as output. Even though CNN performs very well in the image classification task, sometimes the requirement of huge data for more accurate prediction limits its use as a classifier. As a result, in the current work, rather than using the CNN model as a classifier, we only used it as a features extractor.

Figure 4 shows the architecture of the proposed feature extractor. It mainly consists of a CNN having four convolution layers and spatial attention sub-networks. The spatial attention sub-networks, which are variants of widely used CNNs, use attention modules to fine-tune the feature maps in each convolution layer, thereby enhancing CNN's learning ability.

Following each convolution layer, we have used a max-pooling layer to lessen data variance and a dropout layer to avoid over-fitting. Before the max-pooling layer, the attention feature maps from the spatial attention sub-network are added to re-calibrate the original features. This layering scheme is repeated three times with a different number of 3×3 filters. All neurons of these convolution layers have Re-LU (Rectified Linear Unit) as an activation function to learn the nonlinear representation. The details of the network architecture are given in Table 1.

At last, the output features are first flattened and then pass through a fully connected layer, which generates a 1024-dimensional feature vector from the input image.

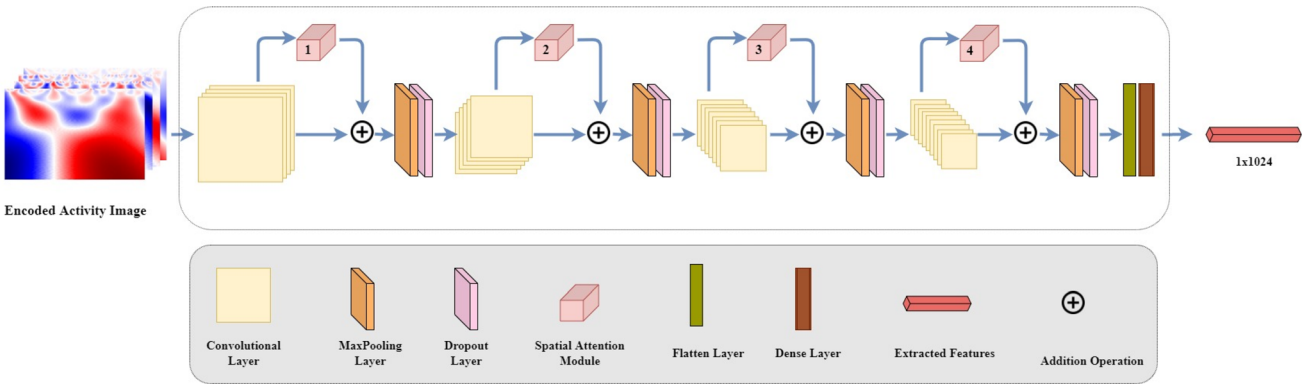

Fig. 4 Architecture of the proposed CNN-based feature extractor

Table 1 Details of the CNN architecture used for the purpose of feature extraction. SAM here refers to Spatial Attention Module

Layer	Type	Filter size	No. of filters	Strider
Input	128 × 128 × 6 for UCI-HAR 128 × 128 × 12 for MHEALTH 80 × 80 × 6 for WISDM 128 × 128 × 27 for PAMAP2 128 × 128 × 3 for HHAR	–	–	–
conv2D_1	Conv2D + ReLU	3x3	32	1x1
SAM_1	–	–	–	–
Max_pooling2D_1	MaxPooling2D	2 × 2	–	–
Dropout_1	Dropout (20%)	–	–	–
Conv2D_2	Conv2D + ReLU	3 × 3	64	1 × 1
SAM_2	–	–	–	–
Max_pooling2D_2	MaxPooling2D	2 × 2		
Dropout_2	Dropout (20%)	–	–	–
Conv2D_3	Conv2D + ReLU	3 × 3	64	1 × 1
SAM_3	–	–	–	–
Max_pooling2D_3	MaxPooling2D	2 × 2	–	–
Dropout_3	Dropout (20%)			
Conv2D_4	Conv2D + ReLU	3 × 3	128	1 × 1
SAM_4	–	–	–	–
Max_pooling2D_4	MaxPooling2D	2 × 2	–	–
Dropout_4	Dropout (20%)	–	–	–
Flatten	Flatten()	–	–	–
Output	Fully Connected Layer (1024 units) + ReLU	–	–	–

3.4 Spatial attention module

Recently, the attention mechanisms attract more and more researchers' interest and have been widely used with the CNN and RNN models in many domains like computer vision and image processing. This mechanism enables the network to pay more focus to some discriminating regions in certain time periods, which improves the learning ability of the network. In this article, we design a class of attention module to focus on where is an informative part present in the encoded images.

The proposed spatial attention module generates a spatial attention feature map by utilizing the inter-spatial relationship of features. As shown in Fig. 5, a 1 × 1 convolution layer is first used to fuse the information along the channels, generating a 2D feature map $Y \epsilon R^{H \times W}$. Then, we apply two 2D convolution layers to generate the spatial attention features map $Y_{sa} \epsilon R^{H \times W \times C}$. For these two 2D convolution layers, the number of convolution filters varies

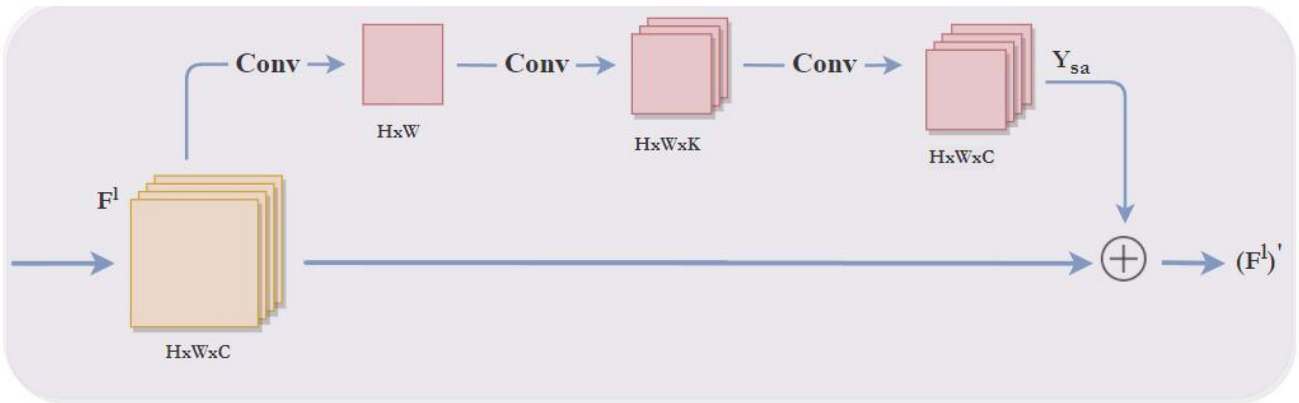

Fig. 5 Illustration of the Spatial Attention Module used in the present work

Table 2 Details of SAM used in the present work. SAM refers to Spatial Attention Module

Module	Convolution Layer-1		Convolution Layer-2		Convolution Layer-3	
	Filter size	No of filters	Filter size	No of filter	Filter size	No of filters
SAM-1	1×1	1	3×3	16	3×3	32
SAM-2	1×1	1	3×3	32	3×3	64
SAM-3	1×1	1	3×3	32	3×3	64
SAM-4	1×1	1	3×3	64	3×3	128

from module to module. The details of spatial attention module architecture are shown in Table 2.

We use Re-LU as the activation function for the convolution layers and padding operator to avoid the change in spatial size. Finally, we use Y_{sa} to re-calibrate F^l using Eq. (5).

$$\left(F^l\right)' = Y_{sa} + F^l \tag{5}$$

where F^l is the features map from the previous convolution layer. This $\left(F^l\right)'$ acts as the input for the next CNN layer in the network.

3.5 Feature selection

Feature extraction using CNN produces a large dimension of features, which needs to be processed by the classifier. Many a times, only a small subset of these features is important. The remaining features are redundant or insignificant and only tend to increase the computational time and space. Moreover, the presence of these redundant features also decreases the classification accuracy. To address this issue, FS has been performed on the set of features obtained from the said CNN model. In the proposed method, we have used GA as the unsupervised FS algorithm, and three different filter methods are used to calculate the fitness of each chromosome in the population of GA.

3.5.1 Filter methods

To calculate the fitness of the individual chromosomes, we rely on three filter-based methods, namely MI, ReliefF, and mRMR.

1. **Mutual Information**: MI [92] is used to measure the nonlinear relations between two random variables. It is used to quantify the quantity of data obtained from a random variable by observing the other random variable. It can be referred to as the reduction in uncertainty of a random variable when the other variable is known. Hence, a high MI value suggests a large reduction in uncertainty while a low value suggests less reduction. It can be calculated using Eq. 6:

$$I(X;Y) = \sum_{y\epsilon Y} \sum_{x\epsilon X} P_{X,Y}(x,y) \cdot \log\left(\frac{P_{X,Y}(x,y)}{P_X(x) \cdot P_Y(y)}\right) \tag{6}$$

where $P_{X,Y}(x,y)$ denotes the joint probability density function of X and Y and the marginal density functions are denoted by $P_X(x)$ and $P_Y(y)$. The similarity the joint distribution $P_{X,Y}(x,y)$ to the product of the factored marginal distributions is determined by MI. It equals zero if and only if two random variables are independent, and higher values indicate greater dependency.

2. **Relief-F**: Relief was proposed by Kira and Rendell [93] for binary class problems by using the Euclidean distance measure. Relief-F algorithm is based on the Relief algorithm, a filter method used in FS. Relief was designed primarily for use in the problems of binary classification with discrete or numerical features. Relief assigns a relative weight/score to each feature and acts as a filter method by eliminating the low-ranked features. The feature score changes according to the detection of feature value differences between neighboring instance pairs. If a difference in feature value is discovered with the same class (a 'hit') in a neighboring instance pair, the feature score falls. On the other hand, if a feature value difference is observed with different class values ('miss') in a neighboring instance pair, the feature score climbs. However, it is limited to only two class problems. An extension of the Relief-F algorithm can be used to solve multi-class problems by searching for k closest misses in each class and averaging their contributions for updating W, weighted by each class's prior probability. In the contribution of weights to each feature, it takes the average of k nearest hits and misses. This k can be adjusted and set based on the dataset in question. Furthermore, Relief-F can handle missing data by employing a conditional probability of feature weights. It is defined by the formula given in Eq. (7).

$$W(X_j) = \frac{1}{nK}\sum_{i=1}^{n}\sum_{l=1}^{K}\left(|x_{i,j} - x_{i,j}^{M_l}| - |x_{i,j} - x_{i,j}^{H_l}|\right) \quad (7)$$

where $x_{i,j}$, $x_{i,j}^{M_l}$ or $x_{i,j}^{H_l}$ denotes the j-th component of sample x_i, its l-th closest Miss $x_i^{M_l}$, or its l-th closest Hit $x_i^{H_l}$, respectively. n is the total number of samples, and K is the number of Misses or Hits considered for each sample.

3. **Minimum redundancy maximum relevance**: mRMR [94] is a filter ranking approach in FS that ranks features according to correlation to the class and itself. Preferably, features with a high correlation with the class (output) and a low correlation between themselves are chosen. For continuous features, correlation with the class (relevance) can be evaluated by the F-statistic values and the correlation between features (redundancy) can be determined using Pearson Correlation Coefficient (PCC) values. A greedy search is applied to select the features one by one as the final goal is to maximize the objective function, which is determined by relevance and redundancy. MID (Mutual Information Difference) and MIQ (Mutual Information Quotient) criteria are the two commonly used types of the objective function which represent the difference between relevance and redundancy, or the quotient of relevance and redundancy. It is calculated using the formula given in Eq. 8

$$score_i(f) = \frac{F(f, target)}{\sum_{s\epsilon f'(i-1)} |\mathrm{corr}(f,s)|/(i-1)} \quad (8)$$

where i is the i-th iteration, f is the feature that is evaluated, F is the F-static, $f'(i-1)$ denotes the features selected until $i-1$ iterations, and $corr$ is Pearson correlation.

3.5.2 Genetic Algorithm: an overview

GA is a popular meta-heuristic evolutionary algorithm which is used for solving complex optimization problems. It is a nature-inspired algorithm with biological features like selection, crossover, and mutation. GA comprises the following steps—initial population creation, parent selection, crossover, mutation, and generation of child chromosomes. Initially, a random population is generated with a finite number of chromosomes, each filled with some random values of fixed length. Parent chromosomes are selected from this set of chromosomes which are further used to create the child chromosomes after performing crossover and mutation. A fitness function is defined to evaluate the fitness of each chromosome. If the fitness values of the child chromosomes surpass the fitness of some existing chromosomes in the current population, they replace the chromosomes having low fitness values. The fitness measures the quality of the represented solution obtained at each iteration. These processes are repeated until the generation of the next set of chromosomes that go through the same selection, crossover, and mutation process, and eventually, the subsequent generations are generated through this method. Individuals with the least fitness die as new generations form, making room for new offspring. This leads to a near optimal solution after a fixed number of iterations. A binary version of GA is used in FS, with each chromosome represented as a vector of '0's and '1's. A '0' indicates that the corresponding feature is not selected, whereas a '1' indicates that the corresponding feature is selected.

3.5.3 Proposed GA variant

GA is one of the oldest and classical evolutionary algorithms, inspired by nature. Over the years, various researchers have utilized this algorithm in the field of FS and optimization. It is proved to be one of the best-known algorithms which provide a near-optimal subset of features from the whole feature space. Exploration and exploitation are performed by the key operators, i.e., crossover and mutation. Numerous modifications have been suggested by various researchers to improve GA and reach the near optimal solution. The mutation in GA is decided by a mutation probability which is quite random in nature. Moreover, the fitness of each candidate solution is determined by a learning algorithm (i.e., a classifier) which is often very time-consuming. Keeping the above facts in mind, we propose a modified version of GA which estimates the fitness of the candidate solutions by calculating the aggregate of three filter-based methods, thereby improving the computational time significantly. Also, instead of random mutation, a different mutation method is proposed which improves the fitness of the individual candidate solution. A multi-point crossover is used and for parent selection is done using Roulette wheel for better exploitation. The pseudocode of the mutation technique is described in Algorithm-3.

Algorithm 3: Pseudocode of the mutation technique

Input: *A binary feature vector (F) of size M*
A score vector of length M containing the average of the three filter methods (score).
Output: *An improved binary feature vector (F') of size M.*
Start
$F' = F$
$total = 0$ ▷ sum of scores of non-zero valued features
$n = 0$ ▷ count of non-zero valued features
for $i \leftarrow 1$ **to** M **do**
  if $F(i) \neq 0$ **then**
    $total = total + score(i), n = n + 1$
  end
end
$avgscore = \frac{total}{n}$
for $i \leftarrow 1$ **to** M **do**
  if $rand(0,1) > mutation\_probability$ **then**
    continue
  end
  if $score(i) > avgscore$ *and* $F(i) = 0$ **then**
    $F'(i) = 1$, update avgscore ▷ Add Feature
  end
  if $score(i) < avgscore$ *and* $F(i) = 1$ **then**
    $F'(i) = 0$, update avgscore ▷ Remove Feature
  end
end
End

3.5.4 Fitness function

Wrapper-based FS methods generally use a learning algorithm (i.e., a classifier) to evaluate the fitness of the chromosomes. Since GA is commonly used a wrapper-based method, it follows the same logic; however, it increases the computational time. To overcome this problem, the usage of classifier is replaced by determining the score of each feature vector (i.e., a chromosome) by the help of filter methods, which aids in assessing the strength of each chromosome in an unsupervised way.

A chromosome is a binary vector with '0' indicating that the feature is to be not taken and '1' indicating that the feature is to be taken. By using the three filter methods, we get a filter value (i.e., a score) corresponding to each feature. The filter value of each feature is the average of the value of the three filter methods. We can say that the feature column with the maximum filter value is most important while the feature with the minimum filter value is least important. Hence, to calculate the score of each individual chromosome, we take the mean of the filter values of all the features which are currently '1.' We have described the pseudo-code of the fitness value calculation in Algorithm-4.

Algorithm 4: Pseudocode of the fitness value calculation

Input: *A chromosome (binary vector) of size M.*
Scores of ReliefF, MI and mRMR.
Output: *The fitness value of the chromosome.*
Start
for $i \leftarrow 1$ **to** M **do**
  $score(i) = \frac{ReliefF(i)+MI(i)+mRMR(i)}{3}$
end
$total = 0$ ▷ sum of scores of non-zero valued features.
$n = 0$ ▷ count of non-zero valued features
for $i \leftarrow 1$ **to** M **do**
  if $chromosome(i) \neq 0$ **then**
    $total = total + score(i), n = n + 1$
  end
end
$fitness = \frac{total}{n}$
End

In FS, we intend to increase the classification accuracy of the problem under consideration and decrease the number of features selected simultaneously. In order to do so, we define a single objective function which estimates the overall fitness of each chromosome (feature subset). This objective function is defined in Eq. 9.

$$\text{Fitness}_{\text{overall}} = \alpha \times F + (1 - \alpha) \times \frac{|F| - |f|}{|F|} \tag{9}$$

where F is the fitness of the chromosome, $\alpha \epsilon [0, 1]$ represents the relative weightage between the fitness value and number of features not selected, $|F|$ is the number of features in the given dataset, and $|f|$ is the number of features in the feature subset.

Since we aim to increase the fitness value and reduce the number of features in the feature subset, our objective is to increase the Fitness_overall value.

4 Experiments and results

We have performed experiments using five popular and publicly available HAR datasets—UCI-HAR, WISDM, MHEALTH, PAMAP2, and HHAR. This section contains information about the datasets used, the performance metrics, and the results obtained.

Table 3 Details about different hyper-parameters used during experimentation

Stage	Hyper-parameter	Hyper-parameter value
Features extraction	Optimizer	Adam
	Learning rate	0.001
	Number of epochs	150/100
	Batch size	32
	Population size	10
Features selection	Crossover probability	0.6
	No. of iteration	20
Classification	K value for K-NN	5

4.1 Model implementation

The proposed model is built using the Keras API and the Tensorflow backend. For the CWT part, we have used PyWavelets [95], an open-source python wavelet transform library. The experiments were performed on a laptop with having AMD Ryzen 7 4800 H (2.90 GHz) processor with 16 GB of RAM and NVIDIA GeForce GTX 1660 Ti GPU with 4 GB of VRAM. The PC is powered by a 64-bit Windows 10 operating system.

The feature extractor model is trained under a supervised learning methodology. We have randomly initialized all the weight and bias used for different layers. Adam optimizer is used, and we have tried to minimize the sparse categorical cross entropy losses. The CNN model is trained for 150 epochs with a batch size of 32. Table 3 summarizes the hyper-parameter details used to tuned our model.

For FS techniques, we have experimented with different values of various hyper-parameters. Finally, for our proposed method with FS, we have used 10 as the population size, the value of crossover probability has been set to 0.6. For the KNN classifier, we have set the k value equal to 5.

4.2 Database description

1. **UCI-HAR** [96]: It is a publicly available benchmark dataset for HAR. The dataset was created by recording activities of daily living (ADL) using the embedded inertial sensors of a waist-mounted smartphone. Each participant in a group of 30 volunteers ranging in age from 19 to 48 years performed six activities: Walking, Walking, Upstairs, Walking, Downstairs, Sitting, Standing, and Laying wearing a Samsung Galaxy S II smartphone on their waist. The experiments have been video-recorded to label the data manually. Activity details and corresponding class distribution are given in Table 4. Total nine features (body acceleration, total acceleration, and angular velocity signals in all *X*, *Y*, *Z*-axis) were captured using the embedded accelerometer and gyroscope at a constant sampling rate of 50 Hz. The raw signals were first pre-processed

Table 4 Activity details of UCI-HAR dataset

Activity	Description	Class distribution (in %)
WALKING	The subject walked outside at a brisk to moderate space at a speed of 4–5 km/h	16.72
WALKING_UPSTAIRS	The subject ascended the staircase to a higher floor at a normal space	14.99
WALKING_DOWNSTAIRS	The subject descended the staircase to a lower floor at a normal space	13.65
SITTING	Sitting in a chair in whatever posture the subject feels comfortable	17.26
STANDING	The subject was motionless and did nothing	18.51
LAYING	The subject did nothing but lay still on a bed	18.87

Table 5 Activity details of WISDM dataset

Activity	Description	Class Distribution (in %)
Downstairs	The subject descended a flight of stairs to a lower floor	9.1
Jogging	Running outside at a pace appropriate for each subject	31.2
Sitting	The subject was seated in a chair in a comfortable position	5.5
Standing	The subject did nothing and remained still	4.4
Upstairs	The subject moved up a floor by climbing a set of steps	11.2
Walking	The subject moved outside in a straight line at a pace of 4–5 km/h	38.6

Table 6 Activity details of MHEALTH dataset

Activity	Description	Class distribution (in %)
climbing stairs	The person moved up a floor by climbing a flight of stairs	8.95
Cycling	The subject was cycling down a public street	8.95
Front elevation of arms	The subject was raising the right hand up to 90 degree	8.58
Jogging	The subject was running outside at a speed of 6–7 km/h	8.95
Jump front & back	First, the subject leaped forward, and then, without turning, leaped back to starting position	3.02
Knees bending	The subject slowly bent both knees and then raise the weight up	8.55
Lying down	The subject didn't move while lying motionless on a bed	8.95
Running	The subject was moving forward at a speed of 9–10 km/h	8.95
Sitting & relaxing	In a relaxed position, the individual was seated in a chair	8.95
Standing still	The subject did nothing and remained still	8.95
Waist bends forward	The subject stands steady and reached out to touch the leg with his/her hands	8.25
Walking	The subject went at a speed of 4–5 km/h in a straight line	8.95

by applying noise filters and then sampled using a fixed-length overlapping sliding window of 2.56 s and 50% overlap (128 readings per window). The dataset was randomly partitioned into two parts, where 70% were used for training and the rest 30% for testing.

2. **WISDM** [97]: This dataset contains data collected through controlled laboratory conditions in Fordham University's Wireless Sensor Data Mining laboratory. The samples were captured using a smartphone-embedded accelerometer, and the data collection process was controlled using an application that was executed on an android smartphone. The experiment was carried out on 36 people, and each performed six activities—Walking, Jogging, Sitting, Standing, Upstairs, and Downstairs with an Android phone in their front leg pocket. Table 5 displays a thorough description of the activities and the corresponding class distribution. The 3-axial accelerometer signals were collected at a constant sampling rate of 20 Hz, i.e., each reading at every 50ms and a total of 20 readings per second. For our proposed work, we first sampled the raw signals using a fixed-length overlapping window of 4 s and 50% overlap (80 readings per window).
3. **MHEALTH** [98, 99]: The Mobile Health (MHEALTH) dataset is a multi-modal wearable sensor dataset. The dataset contained body motion and vital signs recordings for 10 volunteers of diverse profiles. Each volunteer performed 12 different physical activities (Standing Still, Lying Down, Walking, Climbing Stairs, Cycling, Jogging, Running, etc.) wearing three wearable sensors (accelerometer, gyroscope, and 2-led electrocardiogram). The activity details and the class distributions are shown in Table 6. The sensors were attached to the chest, right wrist, and left ankle with elastic straps. Using these sensors, various motions like acceleration, angular velocity, magnetic field orientation were measured for better body dynamics while performing different activities. All the sensor modalities were recorded at a constant sampling rate of 50 Hz. For our proposed work, we consider only the accelerometer and gyroscope sensors readings placed on different body parts. We first sampled the raw signals using a fixed-length overlapping window of 2.56 s and 50% overlap (128 readings per window).
4. **PAMAP2** [100]: The hardware configuration for the PAMAP2 dataset includes three Inertial Measurement Units (IMUs) that are positioned above the wrist of the dominant arm, over the chest, and at the ankle. The dataset has been recorded at a frequency of 100 Hz. The entire set of data includes a class of 9 people with annotated human activities who had specific physical descriptions. Most of the participants were men, and their dominant hand was the right hand. In actuality, PAMAP2 has only one left-handed and one female subject, with ids 102 and 108, respectively. Each individual was required to adhere to a protocol that included 12 separate tasks. A detailed description of all the activities and the class distribution are shown in Table 7. There are almost 10 h of activity data in this collection. After removing the anomalous data, we have segmented the sensor data by a fixed-length sliding window with 50% overlapping. We have then randomly partitioned the dataset into two parts, where 70% are used for training and the remaining 30% for testing.
5. **HHAR** [101]: The Heterogeneity Dataset for Human Activity Recognition (HHAR) from Smartphone and

Table 7 Activity details of PAMAP2 dataset

Activity	Description	Class distribution (in %)
Nordic_walking	The subject performed outside on asphaltic terrain, using asphalt pads on the walking poles	9.68
Ascending_stairs	The subject covered a distance of five floors while going upstairs	6.04
Cycling	The subject was riding a real bicycle with slow to moderate space	8.47
descending_stairs	The subject covered a distance of five floors while going downstairs	5.40
Ironing	The subject was ironing 1–2 shirts or T-shirts	12.28
Lying	The subject was lying quietly while doing nothing, small movements were allowed	9.90
Rope_jumping	The subjects used the method that worked best for them, which was typically the basic leap or the alternate foot jump	2.54
Running	The subject was jogging outside at a suitable speed	5.06
Sitting	The subject was permitted to sit in a chair in whatever position that makes them feel comfortable and to switch positions while they are there	9.54
Standing	The subject was motionless and stood still	9.78
Vacuume_cleaning	The subject was vacuum cleaning one or two office rooms	9.02
Walking	The subject went at a speed of 4–5 km/h in a straight line	12.29

Table 8 Activity details of HHAR dataset

Activity	Description	Class distribution (in %)
Bike	The subject was riding a motorcycle on a free road	16.36
Sit	The subject was lounging comfortably in a chair	17.66
Stairsdown	The subject went down a set of steps to a lower level	14.32
Stairsup	The subject ascended a set of steps to move up a floor	15.80
Stand	The subject showed no action and stood stationary	16.42
Walk	The subject was moving straight ahead at a brisk to moderate speed	19.44

Smartwatches is a dataset used for assessing the performance of various HAR algorithms (classification, automatic data segmentation, sensor fusion, feature extraction, etc.) that use a variety of sensor types. The collection includes readings from two motion sensors, namely accelerometer and gyroscope, frequently found in smartphones, that captured as users carried smartwatches and smartphones while doing some programmed tasks in any sequence. To reflect the sensor heterogeneity which can be anticipated in actual deployments, the dataset is compiled using a variety of device models and use scenarios. This dataset recorded 6 different activities of 9 individuals using 6 types of mobile devices (4 smartphones and 2 smartwatches). Table 8 shows detailed description and class distribution of HHAR dataset. In our experiment, we have used only the smartphone's accelerometer data. We have divided the sensor data into segments using a fixed-length sliding window with 50% overlapping. The dataset is then randomly divided into two sections, with 30% being used for testing and the rest 70% being used for training.

Table 9 presents the summarized information about the five datasets. UCI-HAR, WISDM, and HHAR datasets contain 6 activities, but the number of sensors is different. The MHEALTH and PAMAP2 both datasets contain the 12 activities with more additional sensors. HHAR contains the largest number of training and testing data, whereas PAMAP2 contains more additional sensors compared to the rest of the datasets.

4.3 Performance metrics

In this paper, we mainly use accuracy, precision, recall, F1—score, and confusion matrix as the performance measures. We have used micro-averaging score for calculating precision, recall, and F1—score. Accuracy is defined as the proportion of correctly predicted samples to the total number of samples. A True Positive (TP) outcome is one in which the model correctly predicts the positive class. A True Negative (TN), on the other hand, is an outcome in

Table 9 Details of the datasets used

Dataset	No. of activities	Sensors	Sampling rate (in Hz)	No. of training samples	No. of test samples
UCI-HAR	6	Accelerometer, Gyroscope	50	7352	2947
WISDM	6	Accelerometer	20	5806	1452
MHEALTH	12	Accelerometer, Gyroscope, Magnetometer, 2-led ECG	50	4288	1073
PAMAP2	12	Accelerometer, Gyroscope, Magnetometer	100	21,249	9107
HHAR	6	Accelerometer, Gyroscope	50–200	123,365	52,872

which the model correctly predicts the negative class. Similarly, a False Positive (FP) is an outcome in which the model predicts the positive class incorrectly and a False Negative (FN) is an outcome in which the model predicts the negative class incorrectly. The accuracy can be calculated in terms of TP, TN, FN, and FP using Eq. (10).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

1. **Precision:** Precision is defined as the percentage of positive samples identified correctly, based on the total number of samples identified as positive. Precision can be calculated using Eq. (11).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

2. **Recall:** Recall is the proportion of positive samples that are accurately identified out of all positive trials. We can calculate the recall using Eq. (12).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

3. **F1-score:** F1-score is a comprehensive approximation of the model's accuracy, and it is nothing but the harmonic mean of precision and recall. It can be calculated using Eq. (13).

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

4. **Confusion matrix:** Confusion matrix is a square matrix that represents the overall performance of a classification model. The rows of the confusion matrix represent true class label instances, while the columns represent predicted class label instances. This matrix's diagonal elements count the number of trials where the predicted label equals the true label. The confusion matrix is an important metric for visualizing the model's classification performance.

4.4 Results

To thoroughly measure the performance of the proposed models, we first evaluate the method without FS and compare it with the result found using the method with FS. Table 10 summarizes the performance of our proposed model without FS.

Use of the FS technique helps us reduce the number of features, which also improves the overall accuracy of our model. Table 11 provides the detailed performance metrics obtained by our model using the FS method.

From Tables 10 and 11, it can be seen that the FS technique reduces the size of the feature set by almost 1/3 of the original feature set in the majority of the cases. This reduced feature set improves the recognition accuracy by 0.71% for UCI-HAR, 1.06% for WISDM, 0.18% for MHEALTH, 0.76% for PAMAP2, and 0.88% for HHAR datasets.

The accuracy and loss plots obtained using the feature extractor model on the UCI-HAR dataset are shown in Fig. 6, while the accuracy and loss plots for WISDM, MHEALTH, PAMAP2, and HHAR datasets are shown in Figs. 7, 8, 9, 10, respectively.

Table 10 Performance details of the proposed method without FS

Dataset	No. of extracted features	Accuracy (in %)	Precision	Recall	F-1 score
UCI-HAR	1024	98.74	0.9874	0.9874	0.9874
WISDM	1024	98.34	0.9834	0.9834	0.9834
MHEALTH	1024	99.72	0.9972	0.9972	0.9972
PAMAP2	1024	97.55	0.9755	0.9755	0.9755
HHAR	1024	96.87	0.9687	0.9687	0.9687

Table 11 Performance details of the proposed method with FS

Dataset	Np. of selected features	Accuracy (in %)	Precision	Recall	F-1 score
UCI-HAR	242	99.45	0.9945	0.9945	0.9945
WISDM	380	99.38	0.9938	0.9938	0.9938
MHEALTH	307	99.90	0.9990	0.9990	0.9990
PAMAP2	332	98.29	0.9829	0.9829	0.9829
HHAR	515	97.72	0.9772	0.9772	0.9772

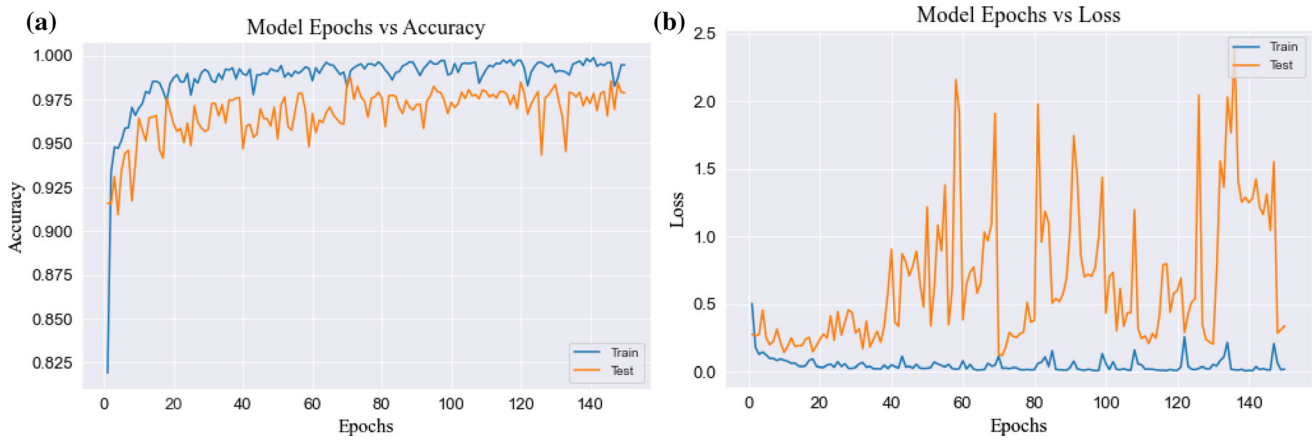

Fig. 6 **a** Accuracy plots for training and testing. **b** Loss plots for training and testing obtained using the feature extractor model on UCI-HAR dataset

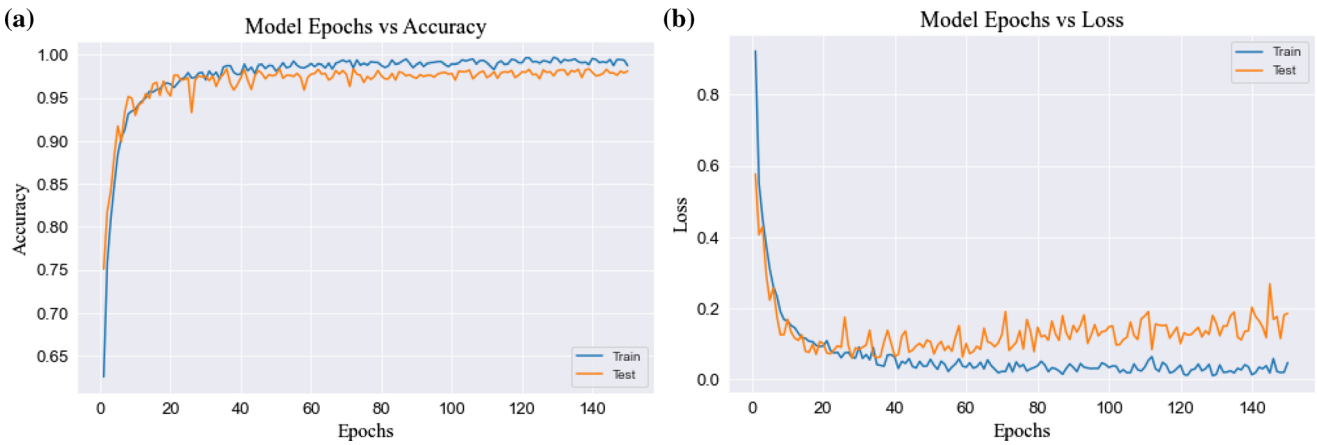

Fig. 7 **a** Accuracy plots for training and testing. **b** Loss plots for training and testing obtained using the feature extractor model on WISDM dataset

4.4.1 Evaluation on UCI-HAR dataset

Figure 11 shows the confusion matrices of the proposed method without FS and with FS side by side.

On the UCI-HAR dataset before applying FS, out of 2947 test samples, a total of 2910 samples are correctly classified by our model. After applying the FS technique, the total number of correctly classified samples increases to 2931, and overall, the accuracy is improved from 98.74 to 99. 45%. If we compare Fig. 11a, b, we can see that FS technique improves the discrimination between Standing and Sitting. It also improves the recognition accuracy of the walking activity class. Even after applying the FS, there is still confusion between sitting and standing. The main reason could be that the two exercises are comparable from the perspective of movement sensors. Data from accelerometers and gyroscopes alone are insufficient for mining dipper discriminative information.

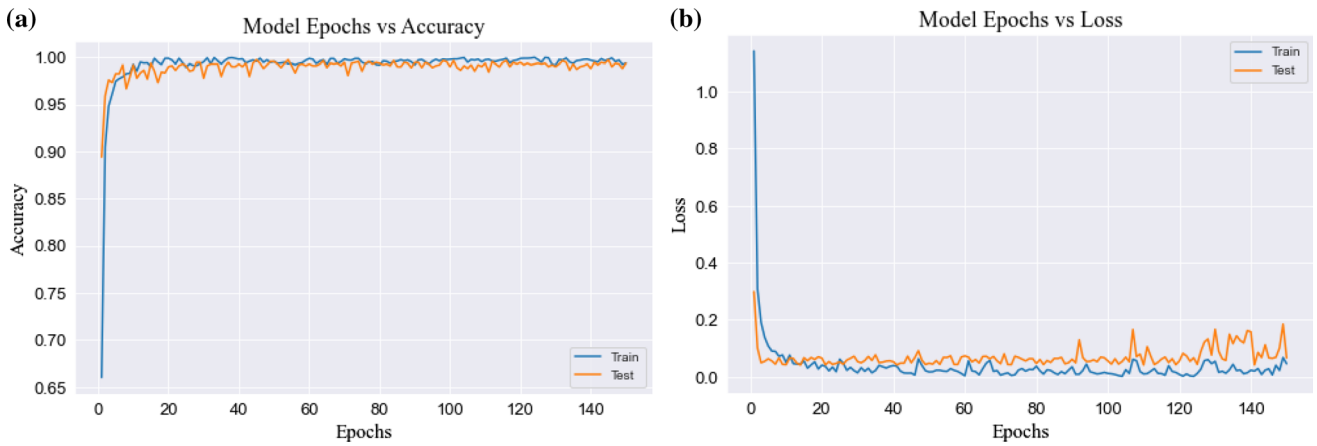

Fig. 8 **a** Accuracy plots for training and testing. **b** Loss plots for training and testing obtained using the feature extractor model on MHEALTH dataset

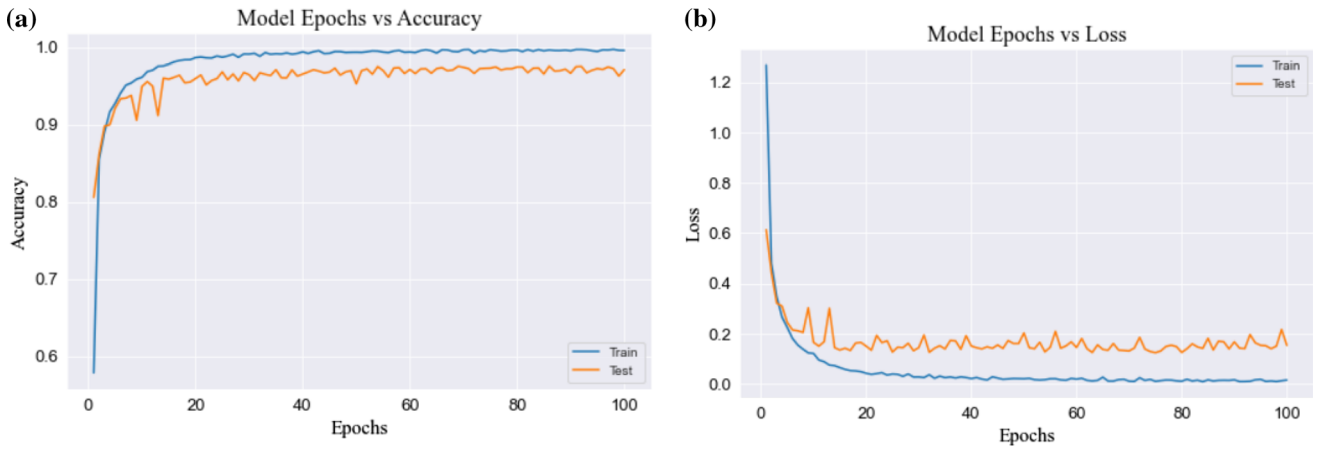

Fig. 9 **a** Accuracy plots for training and testing. **b** Loss plots for training and testing obtained using the feature extractor model on PAMAP2 dataset

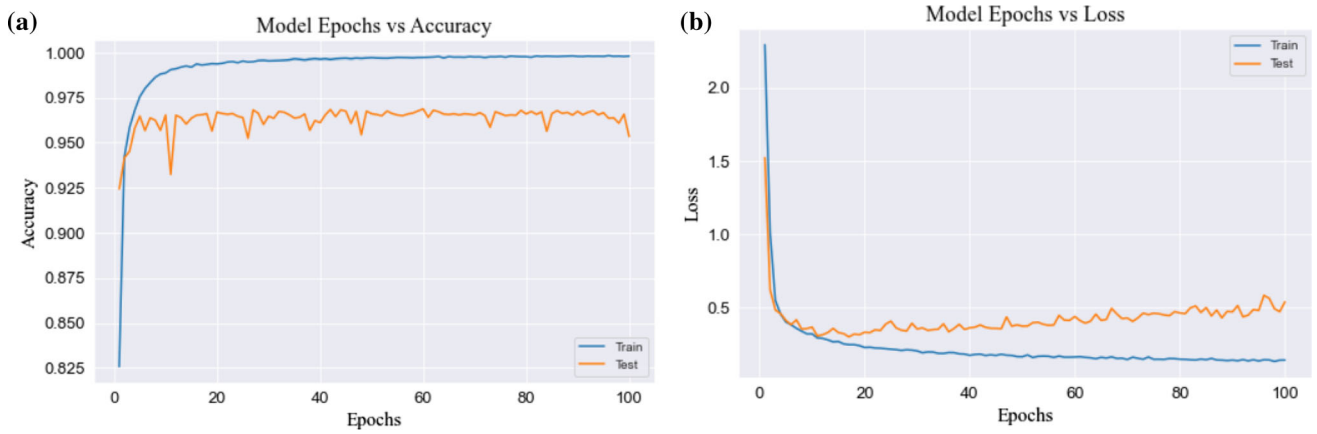

Fig. 10 **a** Accuracy plots for training and testing. **b** Loss plots for training and testing obtained using the feature extractor model on HHAR dataset

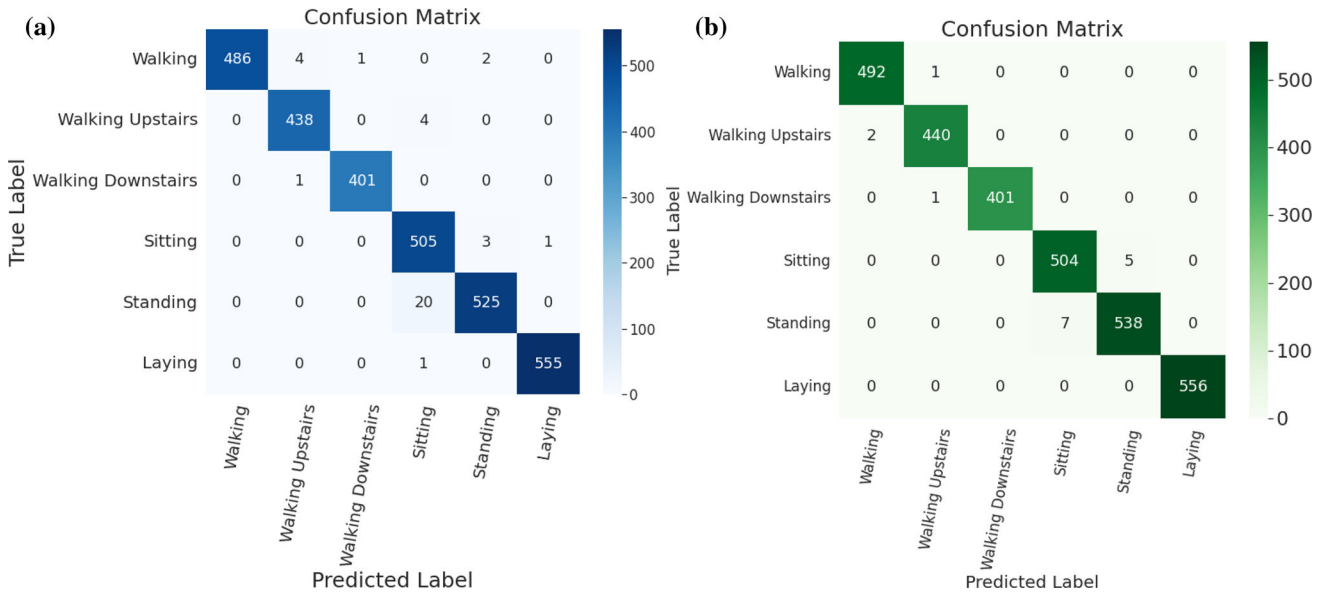

Fig. 11 Confusion matrices for UCI-HAR on the model **a** without FS and **b** with FS

4.4.2 Evaluation on WISDM dataset

When we have tested our trained model on the WISDM dataset, the FS techniques improve the overall recognition accuracy from 98.34 to 99.38%. Figure 12 represents confusion matrices of our proposed method without and with FS. If we compare the confusion matrices of Fig. 12, it is clear that the reduced optimal features map generated by the FS technique helps the classifier to recognize each activity more accurately as the classifier makes less confusion. In the case of WISDM, when we have tested our trained model with 1452 number of new instances, FS techniques increase the number of correctly classified samples from 1428 to 1443.

4.4.3 Evaluation on MHEALTH dataset

In the case of MHEALTH dataset, we have tested our proposed methods with a total of 1052 new samples. Figure 13 depicts the confusion matrices of our proposed method without FS and with FS. The confusion matrices present in Fig. 13 show that though the model without FS performed well, the model gets a little confused while recognizing complex activities like knees bending and

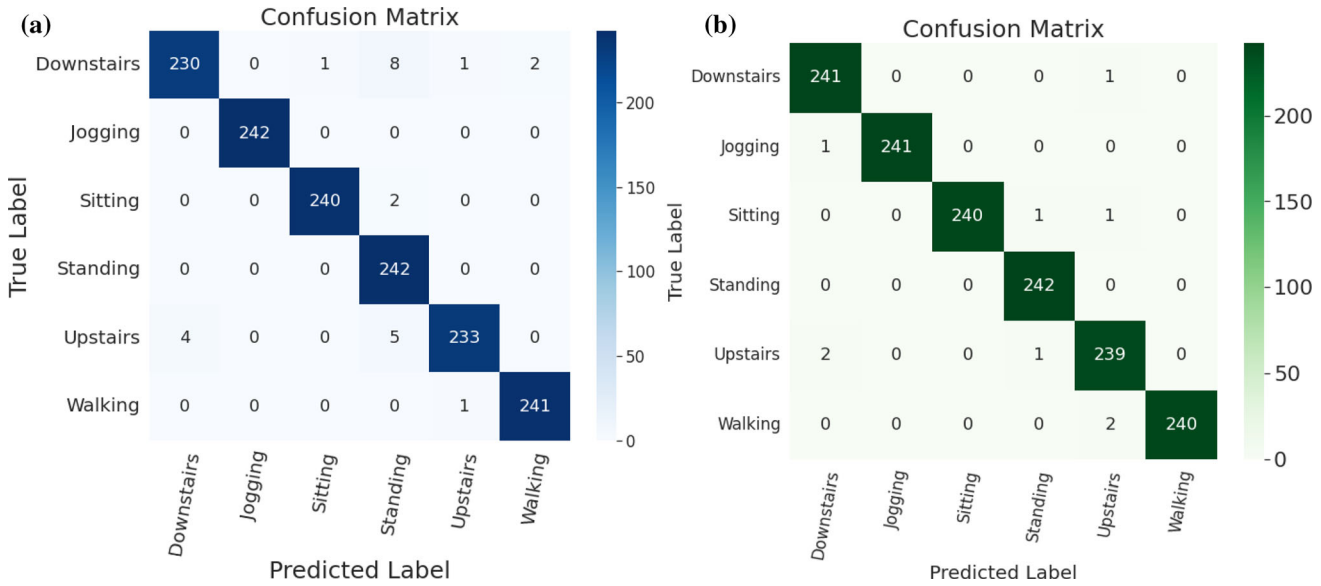

Fig. 12 Confusion matrices for WISDM on the model **a** without FS and **b** with FS

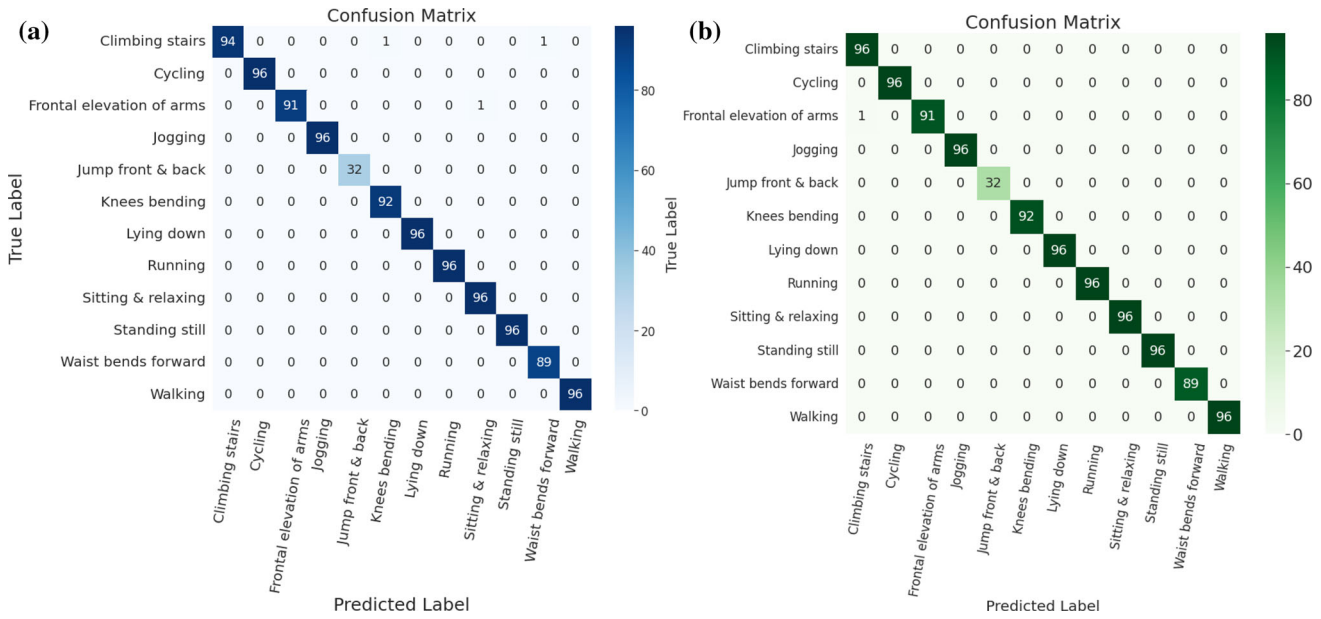

Fig. 13 Confusion matrices for MHEALTH on the model **a** without FS and **b** with FS

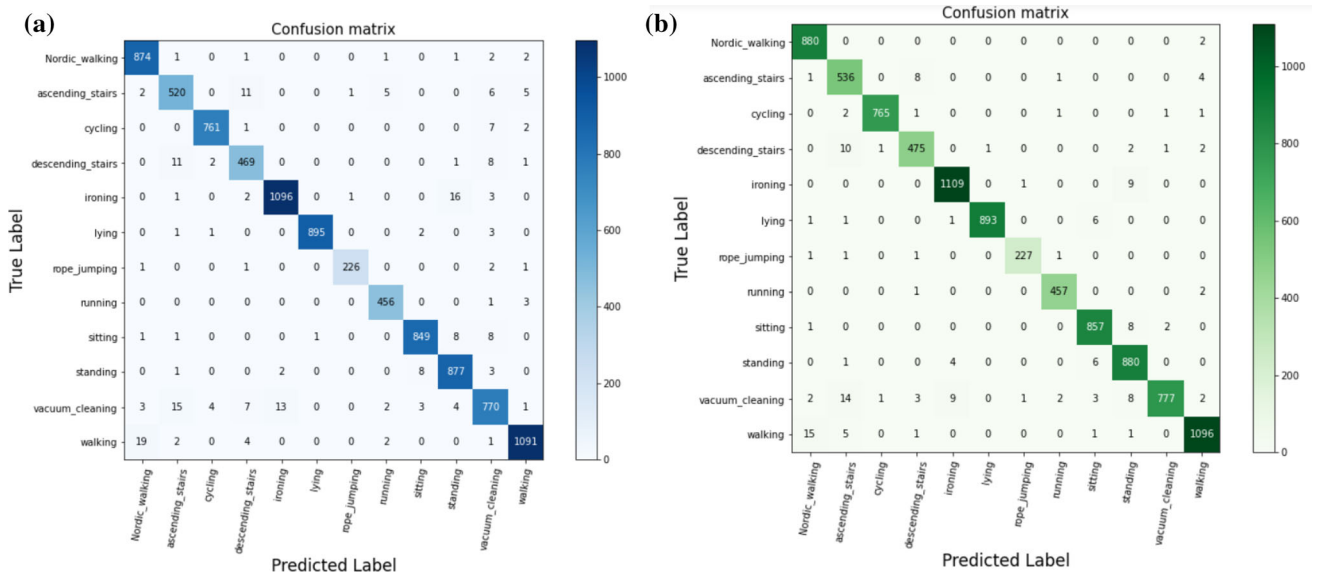

Fig. 14 Confusion matrices for PAMAP2 on the model **a** without FS and **b** with FS

waist bends forward. But FS technique reduces the number of confusions and increases the overall accuracy from 99.72% to 99.90%.

4.4.4 Evaluation on PAMAP2 dataset

The confusion matrices of the proposed technique without FS and with FS are shown side by side in Fig. 14. Prior to using FS, the model obtains 97.55% classification accuracy with a total of 8884 correctly classified samples when tested on a total of 9107 newly created activity samples. With a total of 8952 correctly identified samples, the model achieves 98.29% classification accuracy after applying FS. Even if the FS approach lessens miss-classification, the model still confuses the activity class vacuum_cleaning with other activity classes, as shown in Fig. 14a, b. The complex nature of this activity class is mainly responsible for the confusion.

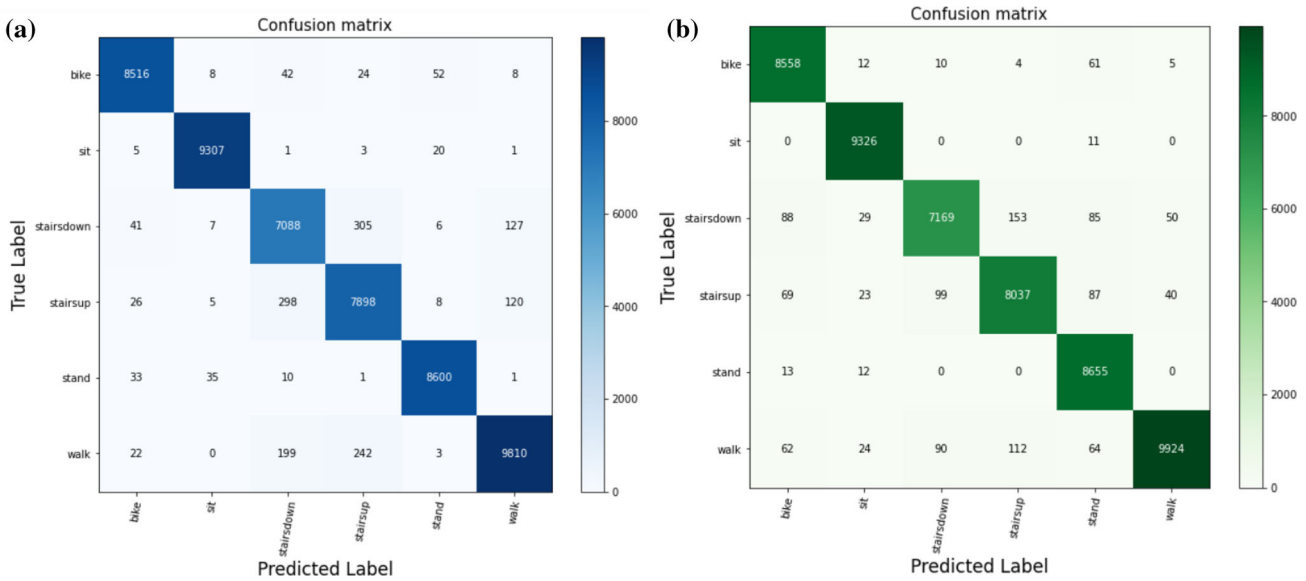

Fig. 15 Confusion matrices for HHAR on the model **a** without FS and **b** with FS

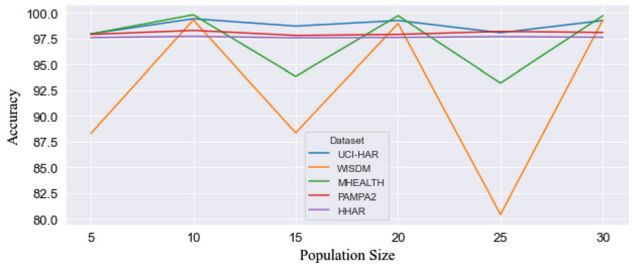

Fig. 16 Population size of GA versus accuracy graphs for all five HAR datasets

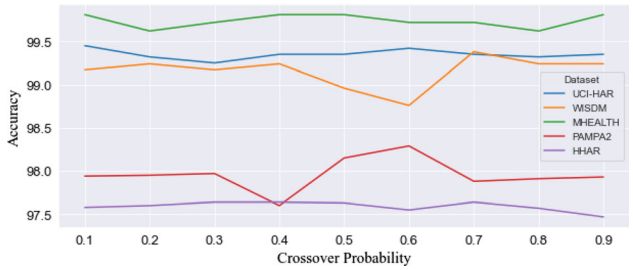

Fig. 17 Crossover probability of GA versus accuracy graphs for all five HAR datasets

4.4.5 Evaluation on HHAR dataset

With a total of 52,872 additional samples used to test our proposed model on the HHAR dataset, the FS approach increases the overall recognition accuracy from 96.87 to 97.72%. The confusion matrices of our suggested technique without and with FS are shown in Fig. 15. We can observe by comparing Figs. 15a, b that the use of the FS approach results in an increase in the number of correctly identified samples from 51,219 to 51,669. Similar to the PAMAP2 dataset, the proposed model still conflates different activity groups even though the FS approach helps to reduce miss-classification. The primary factor may be that the limited accelerometer data from a smartphone may not be sufficient to discern these intricate actions.

4.5 Impact of FS hyper-parameters on model performance

The classification model's performance is greatly influenced by the FS hyper-parameters. This section examines the effect of key FS hyper-parameters such as population size, crossover probability, and the number of iterations on the model's overall accuracy.

4.5.1 Effect of population Size

The population size is an important parameter that has a direct impact on the ability to find the best solution in the search space. Having a large population increases the likelihood of obtaining an optimal solution. In this paper, we have experimented with different population sizes, beginning with 5 and increasing to 30 with a fixed interval of 5. The population size vs accuracy graphs for the five datasets are shown in Fig. 16. For UCI-HAR, WISDM and MHEALTH datasets, the accuracy increases linearly and reaches the global maximum when the population size is 10. As the population size increases, the accuracy follows a zigzag pattern. For WISDM and MHEALTH, accuracy reaches the minimum when the population size is 25. At

the same time, the accuracy does not vary much for PAMAP2 and HHAR datasets. Hence, for our proposed method, we have used 10 as the default population size.

4.5.2 Effect of crossover probability

Crossover is used as a genetic operator for producing new candidate solutions from an existing population stochastically. The crossover probability is the likelihood that a crossover will occur in specific mating. In this experiment, we have varied the crossover probability as 0.1, 0.2 to 0.9 and tried to observe how the accuracy changes. Figure 17 depicts the relation of the crossover probabilities and the accuracy. As we increase the crossover probability, the change in accuracy varies differently for different datasets. For UCI-HAR and MHEALTH datasets, initially, accuracy decreases and then starts to increase as the crossover probability increases. The accuracy reaches the minimum when the crossover probability is 0.3 for UCI-HAR and 0.2 for MHEALTH. For the WISDM dataset, the accuracy first follows a zigzag pattern followed by a sharp fall and reaches the minimum when the crossover probability is 0.6. Further increase in the crossover probability increases the accuracy. The accuracy for the PAMAP2 dataset reaches its lowest point at 0.4 before beginning to rise. The accuracy declines as the crossover probability rises further. On the HHAR dataset, however, the accuracy does not change significantly when the crossover probability rises.

4.5.3 Effect of number of iterations

Figure 18 depicts the change in accuracy as the number of iterations of GA increases. The accuracy of this hyper-parameter, like that of other hyper-parameters, varies depending on the dataset. As we increase the number of iterations from 5 to 30 with a uniform interval of 5, the accuracy of the UCI-HAR dataset gradually increases and reaches a maximum when the number of iterations is 30, whereas for the WISDM and MHEALTH datasets, the accuracy initially increases and then begins to decrease as the number of iterations exceeds 15. When the number of iterations exceeds 25, the accuracy begins to increase again. The accuracy reaches its peak when the number of iterations is set to 10 for the WISDM dataset and 15 for the MHEALTH dataset. With more repetitions, the accuracy for the PAMAP2 dataset grows in a zigzag pattern. In contrast, the accuracy for the HHAR dataset first rises gradually from 5 to 10. The accuracy starts to drop as soon as any iteration is over 10, and it reaches its lowest point at 30. In our experiment, we have used 30 as the default number of iterations.

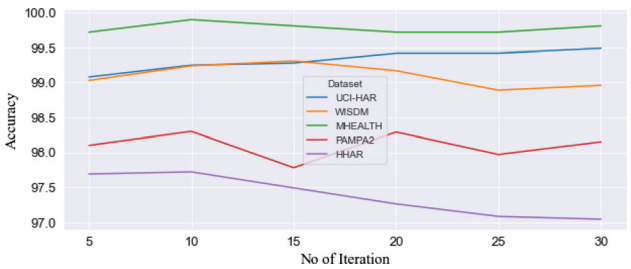

Fig. 18 No. of iteration versus accuracy graphs for all five HAR datasets

4.6 Comparison with state-of-the-art methods

To assess the efficacy and generalizability of our proposed model, we have compared it to a number of state-of-the-art models.

The comparison results for the UCI-HAR, WISDM, MHEALTH, PAMAP2, and HHAR datasets are shown in Tables 12, 13, 14, 15 and 16, respectively. The comparison is done based on the classification accuracy. The results

Table 12 Performance comparison of the proposed model with past methods for the UCI-HAR dataset

Model	Accuracy (in %)
Wang et al [102]	91.65
Nair et al. [28]	94.60
Xia et al. [49]	95.78
Ronao et al. [32]	95.75
Dua et al. [50]	96.20
Challa et al. [103]	96.37
Ignatov et al. [104]	97.63
Proposed model without FS	98.74
Proposed model with FS	**99.45**

Table 13 Performance comparison of the proposed model with past methods for the WISDM dataset

Model	Accuracy (in %)
Sena et al. [105]	89.01
Ignatov et al. [104]	93.32
Lu et al. [106]	93.50
Xia et al. [49]	95.85
Challa et al. [103]	96.05
Mukherjee et al. [107]	97.20
Dua et al. [50]	97.21
Proposed model without FS	98.34
Proposed model with FS	**99.38**

Table 14 Performance comparison of the proposed model with past methods for the MHEALTH dataset

Model	Accuracy (in %)
Chen et al. [108]	94.05
Nguyen et al. [109]	94.72
Lu et al. [106]	96.10
Sena et al. [105]	96.27
Qin et al. [58]	98.50
Uddin et al. [110]	99.00
Abdel et al. [111]	99.68
Proposed model without FS	99.72
Proposed model with FS	**99.90**

Table 15 Performance comparison of the proposed model with past methods for the PAMAP2 dataset

Model	Accuracy (in %)
Xu et al. [112]	93.50
Wang et al. [113]	94.76
Dua et al. [50]	95.27
Awal et al. [114]	95.40
Li et al. [115]	97.37
Baldominos et al. [116]	97.45
Yan et al. [117]	98.18
Proposed model without FS	97.55
Proposed model with FS	**98.29**

Table 16 Performance comparison of the proposed model with past methods for the HHAR dataset

Model	Accuracy (in %)
Lu et al. [118]	91.16
Bai et al. [119]	91.54
Ehatisham et al. [120]	96.10
Gudur et al. [121]	94.88
Qin et al. [58]	95.88
Abdel et al. [111]	97.17
Proposed model without FS	96.87
Proposed model with FS	**97.72**

show that our proposed model without FS has achieved higher recognition accuracy compared to most of the other HAR models. The use of FS technique has improved recognition accuracy even more. For all five datasets, our proposed method with FS outperforms the state-of-the-art algorithms considered here for comparison.

5 Discussion

The overall results shown in the previous section indicate the effectiveness of our proposed models for HAR. The proposed spatial attention module assists in extracting high-quality features by focusing on the specific spatiotemporal properties that the CWT-based encoding is able to express in a better way. In this study, we also analyze how well the FS process works, and we find that in comparison with the initially extracted features, only a limited number of important features are needed for recognizing human activities. In addition to speeding up the computation, the reduced feature set improves recognition accuracy to a significant margin (see Tables 10, 11). These days HAR systems are used in a variety of industries, such as sports analysis, health monitoring, and fall detection for the elderly persons. In sports analysis, the team management needs to analyze players' physical ability and various motion patterns to improve the quality of games. Similarly, in the case of fall detection, an alarm needs to be generated automatically so that a fall may be recognized. Hence, the more accuracy we are able to achieve, the more dependable the system will become. Although our proposed model performs well in most of the cases, it is to be noted that in some cases, it gets confused to distinguish similar activity classes. The model also faces problems to distinguish between activity groups that come with identical sensor data patterns. For example, the model gets confused when sees 'Walking' with 'Upstairs' and 'Downstairs.' Similarly, 'Standing' and 'Sitting' are the most confusing activity classes as both are static activities and generate almost similar signal patterns. Our model finds it difficult to discriminate between "Walking" with "Nordic walking," "Vacuum_cleaning" with "Ironing," and "Upstairs" in the PAMAP2 dataset if we take dataset-specific activities into account. Similar to this, the proposed model for the HHAR dataset frequently conflates the activity classes "stairup" and "stairdown," as well as the activity classes "stairdown" and "walk." Figure 15 shows that following FS, the model misclassifies more "walk" activities as "bike," "sit," and "stand," demonstrating that the FS method does not necessarily decrease the misclassification rate for the confusing cases.

6 Conclusion

Sensor-based HAR deals with the prediction of specific movements or activities of a person based on the sensor data. It has been an interesting research problem as it can be used to obtain the identity of a person, their personality, and psychological state. It can also be applied to identify

complex sport activities and medical domains such as health monitoring systems. Due to its vast scope of practical applications, it is important to ensure that the model fulfills the demanding challenges of the task and hence has gained popularity among the research community in recent times. In this paper, we have proposed a model for HAR based on sensor data. We have used Spatial Attention-aided CNN as the feature extractor and a novel FS technique for selecting the most prominent features using a modified version of the popular evolutionary algorithm called GA. Our proposed method has been experimented on five public datasets—UCI-HAR, WISDM, MHEALTH, PAMAP2, and HHAR. It can be observed that the results obtained are better than state-of-the-art methods. However, there are still some major scopes of improvement to enhance the overall performance of the method. In our future endeavors, we intend to improve the classification accuracy with fewer number of features by exploiting some other recent meta-heuristic algorithms. We also plan to work on some other human activity datasets like video based or still image based and use some pre-trained CNN models to obtain a good set of initial features.

Acknowledgements We are thankful to the Center for Microprocessor Applications for Training Education and Research (CMATER) research laboratory of the Computer Science and Engineering Department, Jadavpur University, Kolkata, India, for providing infrastructural support.

Data Availability Statement All the datasets analyzed during the current study can be downloaded using the following links UCI-HAR—https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones
WISDM—https://www.cis.fordham.edu/wisdm/dataset.php
MHEALTH—http://archive.ics.uci.edu/ml/datasets/mhealth+dataset
PAMAP2—https://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring
HHAR—http://archive.ics.uci.edu/ml/datasets/heterogeneity+activity+recognition.

Declarations

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

References

1. Mabrouk AB, Zagrouba E (2018) Abnormal behavior recognition for intelligent video surveillance systems: a review. Expert Syst Appl 91:480–491
2. Ogbuabor G, La R (2018) Human activity recognition for healthcare using smartphones. In: Proceedings of the 2018 10th international conference on machine learning and computing, pp 41–46
3. Dhiman C, Vishwakarma DK (2019) A review of state-of-the-art techniques for abnormal human activity recognition. Eng Appl Artif Intell 77:21–45
4. Mosquera JH, Loaiza H, Nope SE, Restrepo AD (2017) Identifying facial gestures to emulate a mouse: navigation application on Facebook. IEEE Latin Am Trans 15(1):121–128
5. Roudposhti KK, Dias J, Peixoto P, Metsis V, Nunes U (2016) A multilevel body motion-based human activity analysis methodology. IEEE Trans Cognit Develop Syst 9(1):16–29
6. Banerjee A, Roy S, Kundu R, Singh PK, Bhateja V, Sarkar R (2022) An ensemble approach for still image-based human action recognition. Neural Comput Appl. pp 1–14
7. Chakraborty S, Mondal R, Singh PK, Sarkar R, Bhattacharjee D (2021) Transfer learning with fine tuning for human action recognition from still images. Multimed Tools Appl 80(13):20547–20578
8. Reyes-Ortiz JL, Oneto L, Samà A, Parra X, Anguita D (2016) Transition-aware human activity recognition using smartphones. Neurocomputing 171:754–767
9. Xu H, Huang Z, Wang J, Kang Z (2017) Study on fast human activity recognition based on optimized feature selection. In: 2017 16th international symposium on distributed computing and applications to business, engineering and science (DCABES). IEEE, pp 109–112
10. Nurhanim K, Elamvazuthi I, Izhar L, Ganesan T (2017) Classification of human activity based on smartphone inertial sensor using support vector machine. In: (2017) IEEE 3rd international symposium in robotics and manufacturing automation (ROMA). IEEE, pp 1–5
11. Paul P, George T (2015) An effective approach for human activity recognition on smartphone. In: 2015 IEEE international conference on engineering and technology (ICETECH). IEEE, pp 1–3
12. Sani S, Wiratunga N, Massie S (2017) Learning deep features for kNN-based human activity recognition. In: Proceedings of the ICCBR 2017 Workshops. CEUR Workshop Proceedings
13. Liu Z, Li S, Hao J, Hu J, Pan M (2021) An efficient and fast model reduced kernel knn for human activity recognition. J Adv Transp. 2021
14. Fan L, Wang Z, Wang H (2013) Human activity recognition model based on decision tree. In: 2013 international conference on advanced cloud and big data. IEEE, pp 64–68
15. Brajesh S, Ray I (2020) Ensemble approach for sensor-based human activity recognition. In: Adjunct Proceedings of the 2020 ACM international joint conference on pervasive and ubiquitous computing and proceedings of the 2020 ACM international symposium on wearable computers, pp 296–300
16. Hnoohom N, Mekruksavanich S, Jitpattanakul A (2017) Human activity recognition using triaxial acceleration data from smartphone and ensemble learning. In: 2017 13th international conference on signal-image technology and internet-based systems (SITIS). IEEE, pp 408–412
17. Al-Makhadmeh Z, Tolba A (2020) Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing 102(2):501–522
18. Zhang X, Zhang Y, Hu Q (2019) Deep learning based vein segmentation from susceptibility-weighted images. Computing 101(6):637–652
19. Semwal VB, Mondal K, Nandi GC (2017) Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach. Neural Comput Appl 28(3):565–574
20. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
21. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25, pp 84-90.
22. Wang Z, Oates T(2015) Imaging time-series to improve classification and imputation. In: 24th international joint conference on artificial intelligence

23. Silva DF, De Souza VM, Batista GE (2013) Time series classification using compression distance of recurrence plots. In: 2013 IEEE 13th international conference on data mining. IEEE, pp 687–696
24. Inza I, Larrañaga P, Blanco R, Cerrolaza AJ (2004) Filter versus wrapper gene selection approaches in DNA microarray domains. Artif Intell Med Data Min Genomics Proteomics 31(2):91–103
25. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
26. Singh PK, Kundu S, Adhikary T, Sarkar R, Bhattacharjee D (2021) Progress of human action recognition research in the last ten years: a comprehensive survey. Arch Comput Methods Eng. 29(4), pp 1–41
27. Sarkar A, Banerjee A, Singh PK, Sarkar R (2022) 3D human action recognition: through the eyes of researchers. Expert Syst Appl 193:116424
28. Nair N, Thomas C, Jayagopi DB (2018) Human activity recognition using temporal convolutional network. In: Proceedings of the 5th international Workshop on Sensor-based activity recognition and interaction, pp 1–8
29. Münzner S, Schmidt P, Reiss A, Hanselmann M, Stiefelhagen R, Dürichen R (2017) CNN-based sensor fusion techniques for multimodal human activity recognition. In: Proceedings of the 2017 ACM international symposium on wearable computers, pp 158–165
30. Lee SM, Yoon SM, Cho H (2017) Human activity recognition from accelerometer data using Convolutional Neural Network. In: 2017 IEEE international conference on big data and smart computing (bigcomp). IEEE, pp 131–134
31. Yang J, Nguyen MN, San PP, Li XL, Krishnaswamy S (2015) Deep convolutional neural networks on multichannel time series for human activity recognition. In: 24th international joint conference on artificial intelligence
32. Ronao CA, Cho SB (2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244
33. Huang J, Lin S, Wang N, Dai G, Xie Y, Zhou J (2019) TSE-CNN: a two-stage end-to-end CNN for human activity recognition. IEEE J Biomed Health Inf 24(1):292–299
34. Teng Q, Wang K, Zhang L, He J (2020) The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition. IEEE Sens J 20(13):7265–7274
35. Zhu R, Xiao Z, Li Y, Yang M, Tan Y, Zhou L et al (2019) Efficient human activity recognition solving the confusing activities via deep ensemble learning. IEEE Access 7:75490–75499
36. Zehra N, Azeem SH, Farhan M (2021) Human activity recognition through ensemble learning of multiple convolutional neural networks. In: 2021 55th annual conference on information sciences and systems (CISS). IEEE, pp 1–5
37. Das A, Sil P, Singh PK, Bhateja V, Sarkar R (2020) Mmharensemnet: a multi-modal human activity recognition model. IEEE Sens J 21(10):11569–11576
38. Agarwal P, Alam M (2020) A lightweight deep learning model for human activity recognition on edge devices. Procedia Comput Sci 167:2364–2373
39. Zebin T, Sperrin M, Peek N, Casson AJ (2018) Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks. In: 2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, pp 1–4
40. Liciotti D, Bernardini M, Romeo L, Frontoni E (2020) A sequential deep learning application for recognising human activities in smart homes. Neurocomputing 396:501–513
41. Malshika Welhenge A, Taparugssanagorn A (2019) Human activity classification using long short-term memory network. Signal Image Video Process 13(4):651–656
42. Yu S, Qin L (2018) Human activity recognition with smartphone inertial sensors using bidir-lstm networks. In: 2018 3rd international conference on mechanical, control and computer engineering (ICMCCE). IEEE, pp 219–224
43. Lv M, Xu W, Chen T (2019) A hybrid deep convolutional and recurrent neural network for complex activity recognition using multimodal sensors. Neurocomputing 362:33–40
44. Singh SP, Sharma MK, Lay-Ekuakille A, Gangwar D, Gupta S (2020) Deep ConvLSTM with self-attention for human activity decoding using wearable sensors. IEEE Sens J 21(6):8575–8582
45. Jeyakumar JV, Lee ES, Xia Z, Sandha SS, Tausik N, Srivastava M (2018) Deep convolutional bidirectional LSTM based transportation mode recognition. In: Proceedings of the 2018 ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers, pp 1606–1615
46. Perez-Gamboa S, Sun Q, Improved Zhang Y (2021) Recognition sensor based human activity, via hybrid convolutional and recurrent neural networks. In: 2021 IEEE international symposium on inertial sensors and systems (INERTIAL). IEEE, pp 1–4
47. Mekruksavanich S, Jitpattanakul A (2020) Smartwatch-based human activity recognition using hybrid LSTM network. In: (2020) IEEE SENSORS. IEEE, pp 1–4
48. Mutegeki R, Han DS (2020) A CNN-LSTM approach to human activity recognition. In: 2020 international conference on artificial intelligence in information and communication (ICAIIC). IEEE, pp 362–366
49. Xia K, Huang J, Wang H (2020) LSTM-CNN architecture for human activity recognition. IEEE Access 8:56855–56866
50. Dua N, Singh SN, Semwal VB (2021) Multi-input CNN-GRU based human activity recognition using wearable sensors. Computing 103(7):1461–1478
51. Souza VM, Silva DF, Batista GE (2014) Extracting texture features for time series classification. In: 2014 22nd international conference on pattern recognition. IEEE, pp 1425–1430
52. Garcia-Ceja E, Uddin MZ, Torresen J (2018) Classification of recurrence plots' distance matrices with a convolutional neural network for activity recognition. Procedia Comput sci 130:157–163
53. Hatami N, Gavet Y, Debayle J (2018) Classification of time-series images using deep convolutional neural networks. In: 10th international conference on machine vision (ICMV 2017). International Society for Optics and Photonics, vol 10696, p 106960Y
54. Zhang Y, Hou Y, Zhou S, Ouyang K (2020) Encoding time series as multi-scale signed recurrence plots for classification using fully convolutional networks. Sensors 20(14):3818
55. Hur T, Bang J, Lee J, Kim JI, Lee S et al (2018) Iss2Image: a novel signal-encoding technique for CNN-based human activity recognition. Sensors 18(11):3910
56. Daniel N, Klein I (2021) INIM: inertial images construction with applications to activity recognition. Sensors 21(14):4787
57. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
58. Qin Z, Zhang Y, Meng S, Qin Z, Choo KKR (2020) Imaging and fusing time series for wearable sensor-based human activity recognition. Inf Fusion 53:80–87
59. Ahmad Z, Khan N (2021) Inertial sensor data to image encoding for human action recognition. IEEE Sen J 21(9):10978–10988
60. Ito C, Cao X, Shuzo M, Maeda E (2018) Application of CNN for human activity recognition with FFT spectrogram of acceleration and gyro sensors. In: Proceedings of the 2018 ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers, pp 1503–1510

61. Lawal IA, Bano S (2020) Deep human activity recognition with localisation of wearable sensors. IEEE Access 8:155060–155070
62. Guha R, Khan AH, Singh PK, Sarkar R, Bhattacharjee D (2021) CGA: a new feature selection model for visual human action recognition. Neural Comput Appl 33(10):5267–5286
63. Basak H, Kundu R, Singh PK, Ijaz MF, Woźniak M, Sarkar R (2022) A union of deep learning and swarm-based optimization for 3D human action recognition. Sci Rep 12(1):1–17
64. San Buenaventura CV, Tiglao NMC (2017) Basic Human Activity Recognition based on sensor fusion in smartphones. In: 2017 IFIP/IEEE symposium on integrated network and service management (IM), pp 1182–1185
65. Fan C, Gao F (2021) Enhanced human activity recognition using wearable sensors via a hybrid feature selection method. Sensors (Basel) 21:6434
66. Dewi C, Chen RC (2019) Human activity recognition based on evolution of features selection and random forest. In: 2019 IEEE international conference on systems, man and cybernetics (SMC), pp 2496–2501
67. Nguyen ND, Bui DT, Truong PH, Jeong GM (2018) Position-based feature selection for body sensors regarding daily living activity recognition. J Sens, 2018.
68. Saitoh F (1999) Image contrast enhancement using genetic algorithm. In: IEEE SMC'99 conference proceedings. 1999 ieee international conference on systems, man, and cybernetics (Cat. No.99CH37028). vol. 4, pp 899–904
69. Surya Prabha D, Satheesh Kumar J (2017) An efficient image contrast enhancement algorithm using genetic algorithm and fuzzy intensification operator. Wirel Pers Commun 93(1):223–244. https://doi.org/10.1007/s11277-016-3536-x
70. Arun C, Lakshmi C (2021) Genetic algorithm-based oversampling approach to prune the class imbalance issue in software defect prediction. Soft Comput, pp 1433–7479
71. Ha J, Lee JS (2016) A new under-sampling method using genetic algorithm for imbalanced data classification. In: Proceedings of the 10th international conference on ubiquitous information management and communication. Association for computing machinery. https://doi.org/10.1145/2857546.2857643
72. Sharma DK, Hota HS, Brown K, Handa R (2021) Integration of genetic algorithm with artificial neural network for stock market forecasting. Int J Syst Assur Eng Manag. 13(2), pp 828-841.
73. Chen S, Zhou C (2021) Stock prediction based on genetic algorithm feature selection and long short-term memory neural network. IEEE Access 9:9066–9072
74. Chun DN, Yang HS (1996) Robust image segmentation using genetic algorithm with a fuzzy measure. Pattern Recognit 29(7):1195–1211
75. Phulpagar BD, Kulkarni SC (2011) Image segmentation using genetic algorithm for four gray classes. In: 2011 International conference on energy, automation and signal, pp 1–4
76. Amma NGB (2012) Cardiovascular disease prediction system using genetic algorithm and neural network. In: 2012 international conference on computing, communication and applications, pp 1–5
77. Uyar K, İlhan A (2017) Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks. Procedia Comput Sci 120:588–593. 9th international conference on theory and application of soft computing, computing with words and perception, ICSCCW 2017, 22-23, Budapest, Hungary
78. Hossain S, Mukhopadhyay S, Ray B, Ghosal SK, Sarkar R (2022) A secured image steganography method based on ballot transform and genetic algorithm. Multimed Tools Appl, pp 1–30
79. Khan AH, Sarkar SS, Mali K, Sarkar R (2022) A genetic algorithm based feature selection approach for microstructural image classification. Exp Tech 46(2):335–347
80. Ghosh M, Adhikary S, Ghosh KK, Sardar A, Begum S, Sarkar R (2019) Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods. Med Biol Eng Comput 57(1):159–176
81. Ghosh M, Guha R, Mondal R, Singh PK, Sarkar R, Nasipuri M (2018) Feature selection using histogram-based multi-objective GA for handwritten Devanagari numeral recognition. In: Intelligent engineering informatics. Springer, pp 471–479
82. Malakar S, Ghosh M, Bhowmik S, Sarkar R, Nasipuri M (2020) A GA based hierarchical feature selection approach for handwritten word recognition. Neural Comput Appl 32(7):2533–2552
83. Guha R, Ghosh M, Singh PK, Sarkar R, Nasipuri M (2020) M-HMOGA: a new multi-objective feature selection algorithm for handwritten numeral classification. J Intell Syst 29(1):1453–1467
84. Rostami M, Berahmand K, Forouzandeh S (2021) A novel community detection based genetic algorithm for feature selection. J Big Data. 8(1), pp 1-27.
85. Sharma A, Rani R (2017) An optimized framework for cancer classification using deep learning and genetic algorithm. J Med Imaging Health Inf 12(7):1851–1856
86. Tian H, Chen SC, Shyu ML (2019) Genetic algorithm based deep learning model selection for visual data classification. In: 2019 IEEE 20th international conference on information reuse and integration for data science (IRI), pp 127–134
87. Al-Hatab M, Al-Nima R, Marcantoni I, Porcaro C, Burattini L (2020) Classifying various brain activities by exploiting deep learning techniques and genetic algorithm fusion method. Test Eng Manag 11(83):3035–3052
88. Ghosh M, Guha R, Alam I, Lohariwal P, Jalan D, Sarkar R (2020) Binary genetic swarm optimization: a combination of GA and PSO for feature selection. J Intell Syst 29(1):1598–1610
89. Guha R, Ghosh M, Kapri S, Shaw S, Mutsuddi S, Bhateja V et al (2021) Deluge based genetic algorithm for feature selection. Evolut Intell 14(2):357–367
90. Kilicarslan S, Celik M, Şafak SAHIN (2021) Hybrid models based on genetic algorithm and deep learning algorithms for nutritional Anemia disease classification. Biomed Signal Process Control 63:102231
91. İnce M (2022) Automatic and intelligent content visualization system based on deep learning and genetic algorithm. Neural Comput Appl 34:2473–2493
92. Steuer R, Kurths J, Daub CO, Weise J, Selbig J (2002) The mutual information: detecting and evaluating dependencies between variables. Bioinformatics 18:S231–S240
93. Kira K, Rendell LA (1992) A practical approach to feature selection. In: Sleeman D, Edwards P (eds). Machine learning proceedings 1992. Morgan Kaufmann, pp 249–256. Available from: https://www.sciencedirect.com/science/article/pii/B9781558602472500371
94. Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238
95. Lee G, Gommers R, Waselewski F, Wohlfahrt K, O'Leary A (2019) PyWavelets: a Python package for wavelet analysis. J Open Source Softw 4(36):1237
96. Anguita D, Ghio A, Oneto L, Parra Perez X, Reyes Ortiz JL (2013) A public domain dataset for human activity recognition using smartphones. In: Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning, pp 437–442

97. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. ACM SigKDD Explor Newsl 12(2):74–82
98. Banos O, Garcia R, Holgado-Terriza JA, Damas M, Pomares H, Rojas I et al (2014) mHealthDroid: a novel framework for agile development of mobile health applications. In: International workshop on ambient assisted living. Springer, pp 91–98
99. Banos O, Villalonga C, Garcia R, Saez A, Damas M, Holgado-Terriza JA et al (2015) Design, implementation and validation of a novel open framework for agile development of mobile health applications. Biomed Eng Online 14(2):1–20
100. Reiss A, Stricker D (2012) Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th international symposium on wearable computers. IEEE, pp 108–109
101. Stisen A, Blunck H, Bhattacharya S, Prentow TS, Kjærgaard MB, Dey A et al (2015) Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In: Proceedings of the 13th ACM conference on embedded networked sensor systems, pp 127–140
102. Wang L, Liu R (2020) Human activity recognition based on wearable sensor using hierarchical deep LSTM networks. Circuits Syst Signal Process 39(2):837–856
103. Challa SK, Kumar A, Semwal VB (2021) A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data. Vis Comput, pp 1–15
104. Ignatov A (2018) Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. Appl Soft Comput 62:915–922
105. Sena J, Barreto J, Caetano C, Cramer G, Schwartz WR (2021) Human activity recognition based on smartphone and wearable sensors using multiscale DCNN ensemble. Neurocomputing 444:226–243
106. Lu W, Fan F, Chu J, Jing P, Yuting S (2018) Wearable computing for Internet of Things: a discriminant approach for human activity recognition. IEEE Internet Things J 6(2):2749–2759
107. Mukherjee D, Mondal R, Singh PK, Sarkar R, Bhattacharjee D (2020) EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. Multimed Tools Appl 79(41):31663–31690
108. 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
109. Nguyen H, Tran KP, Zeng X, Koehl L, Tartare G (2019) Wearable sensor data based human activity recognition using machine learning: a new approach. arXiv:1905.03809
110. Uddin MZ, Hassan MM, Alsanad A, Savaglio C (2020) A body sensor data fusion and deep recurrent neural network-based behavior recognition approach for robust healthcare. Inf Fusion 55:105–115
111. Abdel-Basset M, Hawash H, Chang V, Chakrabortty RK, Ryan M (2020) Deep learning for heterogeneous human activity recognition in complex iot applications. IEEE Internet Things J
112. Xu C, Chai D, He J, Zhang X, Duan S (2019) InnoHAR: a deep neural network for complex human activity recognition. IEEE Access 7:9893–9902
113. Wang D, Candinegara E, Hou J, Tan AH, Miao C (2017) Robust human activity recognition using lesser number of wearable sensors. In: 2017 international conference on security, pattern analysis, and cybernetics (SPAC). IEEE, pp 290–295
114. Awal MA, Hasan MK, Rahman MA, Alahe MA et al (2019) Optimization of daily physical activity recognition with feature selection. In: 2019 4th international conference on electrical information and communication technology (EICT). IEEE, pp 1–6
115. Li X, Nie L, Si X, Ding R, Zhan D (2021) Enhancing representation of deep features for sensor-based activity recognition. Mobile Netw Appl 26(1):130–145
116. Baldominos A, Isasi P, Saez Y (2017) Feature selection for physical activity recognition using genetic algorithms. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE, pp 2185–2192
117. Yan Y, Liao T, Zhao J, Wang J, Ma L, Lv W et al (2022) Deep transfer learning with graph neural network for sensor-based human activity recognition. arXiv:2203.07910
118. Lu J, Tong KY (2019) Robust single accelerometer-based activity recognition using modified recurrence plot. IEEE Sens J 19(15):6317–6324
119. Bai L, Yeung C, Efstratiou C, Chikomo M (2019) Motion2Vector: Unsupervised learning in human activity recognition using wrist-sensing data. In: Adjunct proceedings of the 2019 ACM international joint conference on pervasive and ubiquitous computing and proceedings of the 2019 ACM international symposium on wearable computers, pp 537–542
120. Ehatisham-Ul-Haq M, Azam MA, Amin Y, Naeem U (2020) C2FHAR: coarse-to-fine human activity recognition with behavioral context modeling using smart inertial sensors. IEEE Access 8:7731–7747
121. Gudur GK, Sundaramoorthy P, Umaashankar V (2019) Activeharnet: towards on-device deep bayesian active learning for human activity recognition. In: The 3rd international workshop on deep learning for mobile systems and applications, pp 7–12