Personalized Models for Human Activity Recognition with Wearable Sensors: Deep Neural Networks and Signal Processing

Davoud Gholamiangonabadi and Katarina Grolinger

Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada.

*Corresponding author(s). E-mail(s): kgroling@uwo.ca;
Contributing authors: dgholam@uwo.ca;

Abstract

Human Activity Recognition (HAR) has been attracting research attention because of its importance in applications such as health monitoring, assisted living, and active living. In recent years, deep learning, specifically Convolutional Neural Networks (CNNs), have been achieving great results due to their ability to extract features and model complex actions. These generic models work great for the subjects on which they were trained, but their performance degrades substantially for new subjects. Consequently, this paper proposes a personalized HAR model based on CNN and signal decomposition. First, features are extracted from multi-modal sensor data with signal processing techniques, including Stationary Wavelet Transform, Empirical Mode Decomposition (EMD), and Ensemble EMD. Next, CNN carries out the information fusion and the final classification. Personalization is achieved by using a few seconds of the target subject data to select the version of the trained CNN best suited for the target subject. Results show that EMD with cubic spline achieves better accuracy than other signal processing techniques. Moreover, the proposed approach, irrelevant of the type of signal processing, outperforms the state-of-the-art CNN approaches with time-domain features.

Keywords: Human Activity Recognition, Personalized Models, Deep Convolutional Neural Networks, Wearable Sensors, Signal Processing, Sensor Fusion

1 Introduction

Human Activity Recognition (HAR) seeks to detect and identify human activities from the information collected by environmental devices or body-mounted sensors. Environmental devices like cameras require installation and are affected by environmental conditions such as lighting, shadows, and view obstructions, which restricts their application in terms of the monitored spaces. Furthermore, these devices are intrusive and may impose an invasion of privacy [1, 2]. Recent breakthroughs in sensor technologies and developments in Internet of Things (IoT) made data collection from body-mounted sensors straightforward and more economical, resulting in increased proliferation of body area networks. This, together with sensors' robustness, ubiquitousness, and diversity, made multi-sensor fusion with wearable technologies the most common approach for HAR [2, 3]. In contrast to environmental devices, wearable sensors are well suited for multi-user activity recognition in crowded spaces.

A myriad of Machine Learning (ML) techniques have been proposed for HAR, such as K-Nearest Neighbors (KNNs) [4], Support Vector Machines (SVMs) [5], Decision Trees (DTs) [6], Hidden Markov Models (HMMs) [7], and Deep Neural Networks (DNNs) [8]. In recent years, DNNs have been showing great success in HAR, as well as in many other domains including object recognition [9], natural language processing [10, 11], fault detection [12], and emotion recognition [13, 14]. For HAR tasks, DNNs from the category of Convolutional Neural Networks (CNNs) have been dominant [15–18]. CNNs take advantage of spatial relationships within data to extract hierarchical representations of increasing complexity, and then use those representations to perform the ML task.

Sensor-based activity recognition is typically performed with data from different sensors by fusing sensor readings and extracting high-quality features critical for ML methods [19]. Regardless of the fusion approach, irrelevant or redundant features do not contribute to activity recognition and can even degrade the performance while also increasing computational cost [20]. Obtaining a high-quality set of robust features is important for the improvement of accuracy and reduction of computational cost as has been recognized in research studies [21, 22].

Features for multi-sensor fusion can be extracted in time-domain, frequency-domain, or in both [23]. Most HAR studies focus only on the time-domain, and although quite successfully, time-domain approaches may not be sufficient as most human activities generate complex sensor readings. Analyzing data in the frequency-domain can augment time-domain analysis by exposing characteristics of the signal not easily obtainable from the time-domain, such as the presence of different frequencies within the signal and energy distribution over a range of frequencies.

Signal processing techniques, specifically decomposition methods, have been used to improve accuracy in different domains, including geophysical [24], image analysis [25], thermal profile analysis [26], and power quality analysis [27]. According to the type of signal transformation, these techniques can be linear and non-linear. Linear techniques, as their name indicates, are suited for linear systems and examples from this category are Fourier Transforms (FTs) and Wavelet Transforms (WTs). Although FTs have been applied in many domains, they only show what frequencies are present in the signal. In contrast, WTs indicate what frequencies are present and where (or at what scale). In other words, if a signal had a change at some point in time, the FTs would not specify when (time) this has occurred whereas WTs would. Moreover, WTs are fast, computationally light, and maintain temporal information present in the signal [28].

On the other side, non-linear methods such as Empirical Mode Decomposition (EMD) and Ensemble EMD (EEMD) have been utilized for non-linear and non-stationary systems [29]. Although HAR is recognized as a non-linear and non-stationary problem [30], both linear and non-linear approaches have been used [31, 32]. However, those techniques must be examined with respect to how they impact HAR accuracy when used in conjunction with deep learning.

CNNs with signal processing can help fuse information from multiple sensors, but to ensure performance for diverse subjects/people, personalization must be considered. CNNs in HAR work very well when the same person's data is present in the train and the test set; however, the performance drops sharply when the subject in the test is different from the subject(s) in the training set [33]. As we cannot assume that data from each possible subject is available at the time of initial model training, personalized models are required to take into account differences among people. Here, personalization refers to adapting the global model to better suit the specific target subject.

Hence, this paper proposes a personalized model for human activity recognition based on signal processing techniques and deep Convolutional Neural Networks (CNNs). Signal processing techniques first decompose the raw signal from sensors by applying linear and non-linear transformations to create high-quality features. Then, CNN fuses decomposed signals from different sensors to extract higher level features and detect activities. The personalization is achieved by using a few seconds of the target subject's data to dynamically select the version of the trained CNN model best suited for the target subject. Experiments analyze the impact of the linear and non-linear decomposition techniques and examine the effect of personalization. Results show that the signal decomposition improves the HAR accuracy and that using as little as two seconds of the target subject data per activity can improve the model performance.

The rest of the paper is organized as follows: Section 2 explains the background and Section 3 reviews related work. Section 4 describes the methodology, and Section 5 presents the experiments and discusses corresponding results. Finally, Section 6 concludes the paper.

2 Background

This section provides an overview of Wavelet Transform, Empirical Mode Decomposition, and Convolutional Neural Networks.

2.1 Wavelet Transform

The Wavelet Transform (WT) is a signal processing algorithm that introduces a representation of a function (signal) in the time-frequency domain [34, 35] with the help of low-pass and high-pass filters. The signal is decomposed into a set of coefficients, namely approximated and detailed coefficients corresponding to low- and high-frequency bands respectively. A wavelet is a function $\psi \in L^2(R)$ with a zero mean, thus $\int_{-\infty}^{+\infty} \psi(t)dt = 0$.

A Continuous Wavelet Transformation (CWT) represents the continuous signal with two continuous parameters: scale and translation [36]. The scale parameter a defines how stretched/-contracted is the wavelet, while the translation parameter b describes the position in time [37]. The CWT of a signal $x(t)$ at scale a and translation b, $(a, b) \in R$, is defined as follows:

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

where $\psi(t)$ is the mother wavelet, and $*$ indicates the complex conjugate operation.

In practical WT applications, Discrete WT (DWT) and Stationary WT (SWT) are commonly used because of CWT computational complexity [28, 38]. For DWT, equation (1) can be rewritten as:

$$DWT(j,k) = \frac{1}{\sqrt{2^j}} \sum_{-\infty}^{+\infty} x(t)\psi^* \left(\frac{t-k \times 2^j}{2^j} \right) dt \tag{2}$$

where j and k are scale and time shifting, respectively [39].

In DWT and inverse DWT, the critical issues are signal decomposition and reconstruction, respectively. Fig. 1 illustrates N-level DWT decomposition: the process is recursive with low-pass A_i and high-pass D_i filters applied at each decomposition level i. At level L_1, the signal $X(t)$ is decomposed into approximation A_1 and detail D_1 components. Next, at level L_2, approximation A_1 component is decomposed into approximation A_2 and detail D_2 components, and so on. Upon completion, there are one approximation and N detail components.

Stationary Wavelet Transform (SWT) was designed to overcome the lack of translation-invariance of DWT. In SWT, as in DWT, the signal is convolved with low- and high-pass filters, but in SWT there is no decimation. Because of this, in SWT, the signal length is not reduced: the high-frequency and low-frequency components have the same length as the original input. SWT decomposition of a signal $x(n)$ is represented as follows:

$$x(n) = a_j(n) + \sum_{k=1}^{j} d_j(n) \tag{3}$$

$$a_j = \sum_k A_j(n-k)a_{j-1}(k) \tag{4}$$

$$d_j = \sum_k D_j(n-k)d_{j-1}(k) \tag{5}$$

where j is the decomposition level, A_j and D_j are the low-pass and high-pass filters at the level j, k

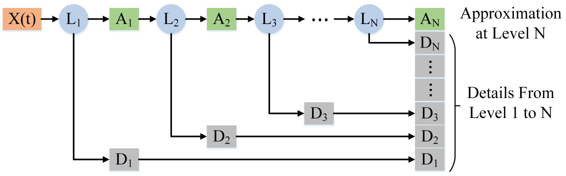


Fig. 1 N-level DWT decomposition based on a recursive filter bank

is the number of mother wavelet coefficients, and n is the sample number [40].

2.2 Empirical Mode Decomposition (EMD)

Huang et al. [41] introduced Empirical Mode Decomposition (EMD), an adaptive time-frequency data analysis technique that can separate a time series into a finite number of components, referred to as Intrinsic Mode Functions (IMFs).

In contrast to wavelet analysis, EMD does not rely on any given wavelet basis and is a more flexible, data-driven approach. EMD iteratively extracts IMFs by means of interpolation between local maxima and minima, and therefore, is well suited for the decomposition of non-linear and non-stationary signals. The initial time series $x(t)$ is decomposed into K IMF components and the final residue r:

$$x(t) = \sum_{k=1}^{K} IMF_k + r \tag{6}$$

A drawback of the EMD technique is a common occurrence of mode mixing, which refers to the situation when a single IMF contains signals of widely disparate scales or when similar scale signals are present in different IMF components [29]. To alleviate this problem, Ensemble Empirical Mode Decomposition (EEMD) has been proposed [29]; EEMD copes with the mode mixing problem by adding white noise to the signal and ensembling EMD trails from signals with different noise.

2.3 Convolutional Neural Network

Artificial Neural Networks (ANNs) mimic the human brain to solve non-linear problems. Similar to the human mind, ANNs learn to perform a task from examples without a need to be explicitly programmed. Convolutional Neural Networks (CNNs), a type of neural networks originally designed for images, are capable of capturing the topology of images, and thus have been greatly successful in image recognition tasks [42]. Later, CNNs have been adapted for non-image applications that can represent data in a grid-like topology; examples include HAR [8, 43] and hand gesture recognition [44].

The CNN architecture consists of layers including convolutional, pulling, and fully connected layers. An example of a CNN with one convolutional, one pooling, and two fully connected layers is shown in Fig. 2. The convolutional layer produces the activation (feature) map by computing the dot product between the learnable filters (kernels) and the output of the previous layer. As the kernel receptive field is small, the network learns the filters that activate on special features present in the input. Next, the pooling layer reduces the data size by down-sampling neuron clusters from the activation maps, consequently reducing CNN computation. In the fully connected layers, similar to Feed Froward Neural Networks (FFNNs), neurons are connected to all nodes in the previous layer. Finally, the output is obtained by applying an activation function; for classification tasks, this is commonly *Softmax* as it represents the output as a probability distribution over classes [45]. During network training, the learnable filters and the weights in fully connected layers are updated by applying backpropagation.

3 Related Work

This section reviews recent works in human activity recognition, signal processing techniques in HAR, and personalization approaches for HAR.

3.1 Human Activity Recognition

In HAR, especially successful are CNN architectures. Ha et al. [46] proposed a CNN with a 2D convolution kernel and a 2D pooling to capture temporal dependencies over time as well as spacial dependencies among sensors. The preprocessing was done with the sliding window technique and the evaluation with the hold-out technique. In experiments with MHEALTH and Skoda datasets,

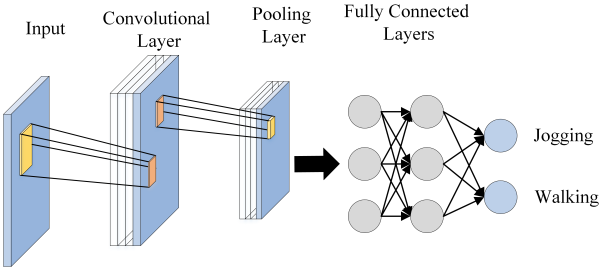


Fig. 2 Convolutional Neural Networks Architecture

their CNN achieved better accuracy than CNN with 1D kernel and k-nearest neighbor with statistical and Fast Fourier Transform (FFT) features.

Uddin and Hassan [47] introduced an activity recognition approach for healthcare based on body sensors and CNNs. Signals from sensors such as ECG, magnetometer, accelerometer, and gyroscope were preprocessed with Gaussian kernel-based Principal Component Analysis (PCA) and Z-score normalization to extract features, and then the CNN was trained with those extracted features. The proposed CNN achieved better accuracy (93.9%) than ANN (87.99%) and Deep Belief Network (90.01%).

Gholamiangonabadi et al. [33] investigated the effectiveness of different preprocessing methods and deep network architectures on HAR accuracy. In experiments with MHEALTH dataset, CNN with a vector magnitude and sliding window technique performed better than other approaches. They compared Leave-One-Subject-Out Cross-Validation (LOSOCV) with traditional k-fold cross-validation and showed the importance of LOSOCV evaluation in HAR. Also, they demonstrated that with a general model, the accuracy changes when activity recognition is done for different target subjects indicating the necessity of personalized models.

Gjoreski et al. [16] proposed an ensemble approach for HAR. First, they carried out a complex preprocessing including windowing, virtual sensor streams, feature engineering in the time- and frequency-domain, and feature reduction. Next, various classical machine learning models such as KNN, SVM, random forest, and Naïve Bayes, as well as deep learning models including CNN and LSTM, were trained. Finally, the trained models were fused into an ensemble with a Hidden Markov Model to capture temporal dependencies. The ensemble outperformed each individual model including deep learning methods.

The main difference between the reviewed studies [16, 33, 46, 47] and ours is that they consider a general, single model for all subjects while we propose a personalization technique to improve HAR accuracy. Gholamiangonabadi *et. al.* [33] considered different DL architectures for improving HAR accuracy and examined evaluation techniques, while our study uses signal processing techniques with DL and proposes a novel personalization approach to improve the accuracy.

3.2 Signal Processing Techniques in Human Activity Recognition

Signal processing techniques can extract high-quality features from time series data and therefore have been used in HAR. In the work of Hanai et al. [48], the features were extracted from the signals with two one-dimensional Haar-like filtering techniques, and then the decision tree (C4.5) was used as a classifier. The evaluation was conducted with data from four subjects and five physical activities by applying 10-fold cross-validation. The results showed that not only is the computational cost for this feature extraction low but also, the achieved accuracy is high.

Xu et al. [32] used Hilbert-Huang transform, EMD, and a feedforward neural network for HAR with PAMAP2 dataset. The features included instantaneous amplitude and frequency derived with EMD, as well as instantaneous energy density and marginal spectrum extracted with Hilbert spectral analysis. They investigated the effect of a multi-feature set versus a single-feature: the results showed that using all four mentioned features improves the performance.

He and Jin [49] presented a hybrid approach based on Discrete Cosine Transform (DCT), PCA, and Multi-class Support Vector Machine (MSVM). First, useful features were extracted from accelerometer data with DCT, then PCA was applied to reduce the number of features, and finally, MSVM was trained with reduced features. The proposed feature extraction, DCT with PCA, performed better than other methods such as wavelet transform, FFT, and statistical features.

Assam and Seidl [50] combined dyadic wavelets, Vector Quantization (VQ), and Hidden Markov Model (HMM) for human activity recognition. The spectral features were extracted from the smartphone's accelerometer signals with multi-resolution wavelet transform, and then the VQ was applied to create codes for HMM. Traditional cross-validation reported 95% accuracy.

Wavelets were also used in several other studies [20, 28, 31]. He et al. [20] proposed a wavelet tensor fuzzy clustering: the Discrete Wavelet Packet Transform (DWPT) method first built feature tensors from activity signals, and then the Multi-linear PCA (MPCA) reduced dimensionality. Finally, activities were recognized with fuzzy clustering. On a dataset containing seven activities from eight subjects, their approach was better than other fuzzy clustering techniques.

Abdu-Aguye and Gomaa [28] decomposed the signal with wavelet transform and then applied an adaptive pooling operator to obtain a compact representation for training a deep CNN. In the evaluation, three different levels of decompositions were considered, namely, three, five, and seven. Experiments with seven different datasets showed that their approach works well on both small and large datasets.

Tian et al. [31] first extracted wavelet energy spectrum features for each activity, and then selected features with an ensemble-based technique. Two classifiers were considered: k-nearest neighbor and support vector machine. Experiments with five subjects and nine activities showed that wavelet energy spectrum features dramatically improve the classifier accuracy.

While [32] used a non-linear method for feature extraction, all other reviewed studies [20, 28, 31, 48–50] applied linear techniques. In contrast, we consider both, linear and non-linear approaches, and compare their performance. Moreover, none of the reviewed studies investigated if there is a statistically significant difference between results, while we check the difference with a statistical test. To further improve the accuracy, we also add the personalization technique.

3.3 Personalization techniques for Human Activity Recognition

In HAR, one model fits all approach suffers from poor generalization as movements are greatly affected by personal characteristics such as height and physical abilities [51]. Personalization aims to adjust the model to better fit the target subject and, consequently, improve recognition accuracy.

Vu and Fujinami [52] proposed a compatibility-based classifier for HAR personalization. Several classifiers such as Random Forest were trained on a set of subjects, and then the most compatible classifier for the target subject was selected. The compatibility was determined by evaluating each of the trained classifiers on a part of the target subject data and choosing the one with the highest F-measure. Evaluation with two public data sets, PAMAP2 and WISDM, has shown that the proposed method improved activity recognition.

To achieve personalization, Ferrari et al. [51] determined the similarity between the target subject and those in the training set based on intrinsic characteristics of signals and physical attributes such as age and weight. Based on the calculated similarities, they put more weight on data from people more similar to the target person. Three approaches for finding the similarity were considered: based on physical features, the signal distance, and the hybrid of the two. To classify activities, three methods were used: Adaboost classifier with handcrafted features, Adaboost with deep features, and SVM with deep features. The results showed that, on average, using the personalized models improves the accuracy.

Lin and Marculescu [53] proposed a personalized model and a training algorithm for HAR. In the personalization model, they combined the Nearest-Class Mean classifiers (NCM) and cosine similarity distance. In the training algorithm, feature extraction was designed based on an optimization function. They evaluated the approach on three public datasets, namely, Sport and Daily Activity, Opportunity, and Sensor Activity. The results showed that the personalized model improved accuracy by only using some instances from the test subject.

While the approach proposed by Ferrari et al. [51] used users' physical characteristics, our approach does not need personal characteristics and solely relies on sensor signals. Similar to Lin and Marculescu [53], we use a part of the target subject data for model personalization. While they update the pre-trained models with the target subject data, we use this data to dynamically select the best model from the training phase. Vu and Fujinami [52] had a similar idea of selecting the best model based on the samples from the target; however, they select the best model from a number of different trained models while we select the best version (parameters) of the same model. In addition to a different personalization technique than those reviewed, our study also differs from the aforementioned papers as it combines signal processing, deep learning, and personalization.

4 Methodology

This paper proposes an approach for personalized HAR based on convolutional neural networks and signal decomposition techniques. CNNs have been selected because of their strong generalization capabilities, abilities to extract features, and recent success in HAR [51]. Signal decomposition techniques are added to extract frequency- and time-domain features and improve accuracy. Finally, personalization further improves accuracy by selecting the model version best suited for the target subject.

This section first describes data preprocessing followed by the personalization approach and the CNN architecture. Finally, the evaluation technique is described.

4.1 Data Preprocessing

For data preparation, two main categories of decomposition approaches are considered: linear techniques (SWT) and non-linear techniques (EMD and EEMD). Techniques are applied individually for each sensor and each recorded feature (e.g., independently for each axis of a 3-axis accelerometer). Table 1 shows the considered techniques while the details are described in the following subsections.

4.1.1 Linear Techniques

Stationary Wavelet Transformation (SWT) is a linear decomposition technique: the original sequence is decomposed into two parts detailed (high-frequency) sequence and approximated (low-frequency) sequence, both of the same length. Then the process continues by decomposing the approximated sequence from the previous level. SWT was selected over DWT because it overcomes the lack of translation invariance of the DWT.

Table 1 Decomposition Techniques

Method	Type	Linear	Non-linear
SWT	db1	✓	
SWT	db2	✓	
SWT	db3	✓	
SWT	db4	✓	
EMD	Cubic Spline		✓
EMD	Linear Spline		✓
EEMD			✓

The wavelet family derived by Daubechies covers the field of orthonormal wavelets; it is extensive and it includes members from extremely localized to very smooth. This study considers four different Mother Wavelets (MWs); db1, db2, db3, and db4, and investigates their effect on HAR. For each db, two scenarios and three decomposition levels are explored as shown in Table 2. In this table, A, D, and LoD stand for the approximation coefficients, detail coefficients, and level of decomposition, respectively. Three levels of decomposition (1, 3, and 5) are designed to explore the effect of increasing decomposition levels.

Scenario F (full) includes all extracted components in the current level, plus previously extracted detailed components while Scenario R (reduced) only uses the approximation coefficients from the current level and previously extracted detailed components. The two scenarios are designed to investigate if the high-frequency component at the current level can be eliminated without losing accuracy: this would simplify DDN models and reduce computational cost.

4.1.2 Non-linear Techniques

Two non-linear decomposition techniques are considered: Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD). In EMD, IMFs are extracted with a sifting method as follows:

1. Identify all the local extrema and connect all local maxima (minima) with a cubic or linear spline to create the upper (lower) envelope.
2. Obtain the first component c by taking the difference between the signal and the local mean of the two envelopes.
3. Consider c as the data. Repeat steps 1 and 2 until c becomes a monotonic function or a function with only one extremum and no more IMFs can be extracted. The final c is designated as residual r.

In other words, EMD decomposes signal $x(t)$ into IMFs c_j and a residual component r_n. The relationship between the original signal, its IMFs, and a residual is expressed as follows:

$$x(t) = \sum_{j=1}^{n} c_j + r_n \tag{7}$$

Table 2 Wavelet Decomposition Scenarios

LoD	Scenario F	Scenario R
1	A1 and D1	A1
3	A3, D3, D2, D1	A3, D2, D1
5	A5, D5, D4, D3, D2, D1	A5, D4, D3, D2, D1

Table 3 Scenarios for EMD and EEMD

Ascending Scenario	Descending Scenario
IMF 1	IMF 6
IMF 1 and 2	IMF 6 and 5
IMF 1, 2 and 3	IMF 6, 5 and 4
IMF 1, 2, 3 and 4	IMF 6, 5, 4 and 3
IMF 1, 2, 3, 4 and 5	IMF 6, 5, 4, 3 and 2
IMF 1, 2, 3, 4, 5 and 6	IMF 6, 5, 4, 3, 2 and 1

As the decomposition is based on the local characteristics of data, this method can be applied to non-stationary and non-linear processes.

Ensemble EMD (EEMD) has been introduced to address the problem of mode mixing present in EMD [29]. EEMD repeatedly adds Gaussian white noises to the original signal and then applies EMD. The process can be summarized with the following steps:

1. Add white noise to the original signal.
2. Decompose the noise-added signal into IMFs by applying the sifting process.
3. Repeat steps 1 and 2 several times, each time with different white noise.
4. Obtain the final IMFs for the ensemble as a mean of IMFs from all repetitions.

As shown in Table 3, for both EMD and EEMD, up to six decomposed IMFs are considered for each original attribute. As with SWT, two scenarios are designed to investigate the impact of the IMF selection on the HAR performance. In Ascending Scenario (AS), IMFs are added one by one starting from IMF 1 to IMF 6 while in Descending Scenario (DS), IMFs are added from IMF 6 to IMF 1.

4.2 Deep Learning Model

The deep learning model used in this study is CNN with the architecture shown in Fig. 3. The sequence of layers is convolutional, max-pooling, convolutional, max-pooling, and two fully connected layers. Each convolutional layer has 64 feature maps, each max-pooling layer has 32 feature maps, and the two fully connected layers have 64 and 32 neurons, respectively. This is the same architecture as CNN-2C-1D presented by [33]: 2C indicates two convolutional layers and 1D refers to a one-dimensional kernel. This architecture was selected as it achieved better accuracy than other CNN and FFNN architectures [33]. The proposed signal processing and personalization techniques could also be applied with other neural networks.

The number of features used for CNN depends on the decomposition type. In the linear approach (SWT), the decomposition level and scenario determine the number of features. For example, Scenario F decomposes 21 original features into 42 features when LoD is one and into 84 when LoD is three.

For the non-linear decomposition, the number of features used for CNN depends on the number of IMF(s). For instance, IMF 1 ascending scenario and IMF 6 descending scenario, both use only one IMF, thus, with 21 original features, the number of features after decomposition remains 21. With two IMFs, IMFs 1 and 2 in ascending or IMFs 5 and 6 in the descending scenario, the number of features is 42.

4.3 Personalization Approach

Natural differences between people's motion patterns can cause low performance of a generic model when applied to different subjects. Personalization techniques aim to remedy this by adapting the model to the specific subject. Here we take advantage of the CNN characteristic to carry out the personalization.

Fig. 4 shows CNN accuracy on the train and test (target) sets with the test set consisting of a single target subject whose data are not in the training set. It can be seen that the training accuracy converges very fast to almost 100% accuracy, but for the target (test) subject, the accuracy fluctuates greatly through epochs. The reason for this is that the CNN is being optimized for the subjects in the training set causing the oscillations for a subject not present in the training set. The main idea is to select the CNN version corresponding to the highest accuracy for the target subject.

The proposed personalization approach is illustrated in Fig. 5. Target subject data and training subjects data undergo signal decomposition separately. The CNN is then trained on the decomposed training data for n epochs. While in traditional CNN training, only the final model is retained, here the models corresponding to each

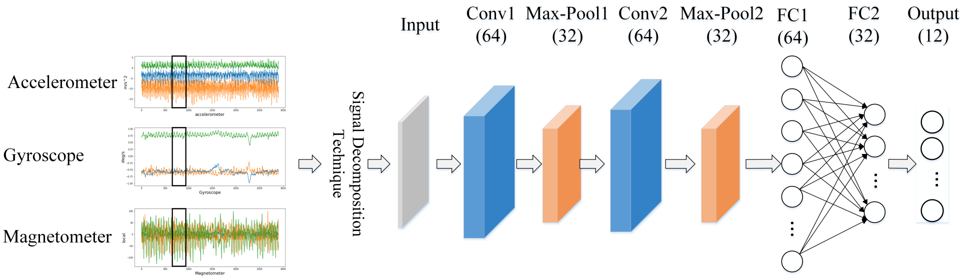


Fig. 3 CNN Architecture

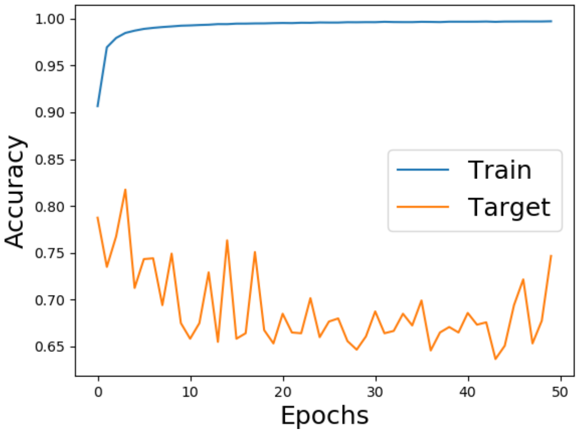


Fig. 4 Train and Test Set Accuracy through the Training Process

epoch (iteration step) are retained for use in the personalization.

The decomposed target subject data is divided into two parts: the validation and the test dataset. The validation set contains an equal number of samples from each activity to ensure personalization across all activities. The duration of activity recording used as a validation set for model personalization is referred to as *personalization time*. Personalization is inspired by the concept of early stopping when the network is iteratively trained on the training set until the error on the validation set starts to increase. In our approach, the validation set is from the target person; therefore, the validation set is used to select the CNN weights from the set of models corresponding to epochs.

In the early stopping, the training terminates when the validation error increases while in our approach, all epochs are processed irrelevant of the validation error. Then, as shown in Fig. 5, our approach selects the best CNN epoch based on the accuracy achieved with the validation set containing the target subject data. This corresponds to selecting the epoch with the highest accuracy in Fig. 4. The weights from this best epoch make the final model for the target subject, and then the model is evaluated on the remaining data from the target user (testing data).

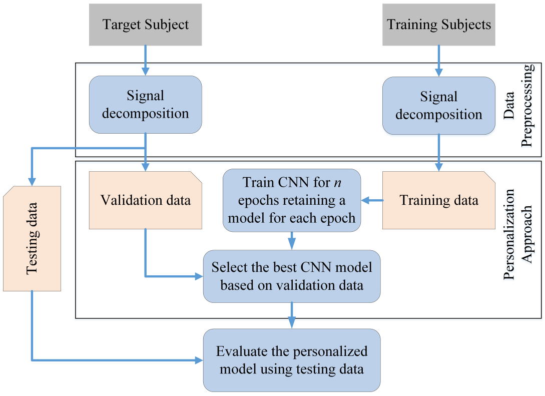

Target Subject
Training Subjects
Signal decomposition
Signal decomposition
Data Preprocessing
Testing data
Validation data
Train CNN for *n* epochs retaining a model for each epoch
Training data
Select the best CNN model based on validation data
Personalization Approach
Evaluate the personalized model using testing data

Fig. 5 Proposed Personalization Technique

An example of training on nine subjects and personalizing the model for the tenth subject is seen in Fig. 4. The train line indicates the accuracy obtained through epochs on the nine training subjects. For personalization, a small fragment of the target subject data is used to select the best model. The models corresponding to each epoch are evaluated with the piece of target subject data as shown with the target line in the figure. In this example, the best accuracy for the target subject is achieved with the model from the third epoch; thus, the model from the third epoch is used for this target subject.

4.4 Evaluation Process and Performance Criteria

Traditional approaches for assessing machine learning models such as hold-out validation and k-fold cross-validation evaluate the model on the samples that did not participate in training. In these approaches, samples for both training and testing come from the same dataset and the selection is typically random. In HAR, data from all subjects are not available at the start of the training and, therefore, the models must be easily adaptable for new subjects.

Leave-One-Subject-Out Cross-Validation (LOSOCV) is an evaluation technique suitable for assessing the performance of generic HAR models on new subjects [33]. In LOSOCV, one subject, referred to as the target subject, is reserved for the evaluation, and the model is trained on the remaining subjects. The process is repeated, each time with a different target subject. However, when the model is personalized with a part of the target subject data, the samples used for the personalization cannot be included in the evaluation. In other words, a small part of the target subject data makes the validation set used for personalizing the model, and the remaining part makes the test set for the final evaluation. This slightly changes LOSOCV to accommodate the personalization technique, but the main LOSOCV process remains the same in that the test samples are not employed in training and personalization.

Accuracy is used as a metric for comparison among models: it assesses the proportion of the samples correctly classified. When the dataset is balanced or approximately balanced, as is the case with our dataset, accuracy is a suitable metric for evaluation [33]. Accuracy is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

where TP, TN, FP, and FN are True Positive, True Negative, False Positive, and False Negative, respectively.

The validation dataset serves the purpose of model personalization and, therefore, its size affects finding the best CNN weights for the target person. A large validation set could potentially improve accuracy, but it also leads to an impractical solution as the majority of the target subject data is then used for building/personalizing the model. Therefore, we aim to employ as small as possible validation set without significant reduction of accuracy. The validation set contains an equal number of samples from all activities. To examine the impact of the validation dataset size, experiments conducted in this study examine the validation sets containing samples from 2, 4, 6, 8, and 10 seconds of recordings per activity.

4.5 Evaluation

This section introduces the dataset and experiments.

4.5.1 Data

The proposed approach was evaluated with MHEALTH (Mobile Health) dataset [54]. The MHEALTH dataset comprises of body motion and vital sign recordings from ten individuals while performing twelve activities including walking, jogging, running, sitting, and others. Three types of sensors, accelerometer, gyroscope, and magnetometer, measured movements of different body parts, namely, acceleration, rate of turn, and magnetic field orientation. Each of the three sensors was placed on the subjects' right wrist and left ankle with an additional accelerometer located on the chest. EEG signal was also recorded in MHEALTH dataset, but it is not used here as it is not deemed to directly relate to short term motion [55].

Each of the sensors records three values corresponding to three axes. Seven sensors used in this study, one on the chest and three sensors on the arm and leg, result in 21 recorded features for each time step. The sampling rate for all sensing modalities is 50 Hz. All 12 recorded activities are used in experiments and reported results are for all activities together.

4.5.2 Experiments

Different scenarios and levels of decomposition were designed to assess the effects of various preprocessing techniques.

For the linear decomposition, SWT method, the following options were considered:

- Three Levels of Decomposition (LoD): 1, 3, and 5

- Two scenarios as discussed in subsection 4.1.1: Scenario Full (Scenario F/SF) and Scenario Reduced (Scenario R/SR)
- Four Mother Wavelets (MWs): db1, db2, db3, and db4
- Five time durations for the validation dataset: 2, 4, 6, 8, and 10 seconds

This makes a total of $3 * 2 * 4 * 5 = 120$ combinations and corresponding experiments for SWT. Table 4 shows the number of features for each scenario and LoD. For example, in LoD 1 and Scenario F, decomposition results in 42 features while in Scenario R, it results in 21 features.

For the non-linear techniques, the following options were considered:

- Three methods: EMD with cubic spline, EMD with linear spline, and EEMD
- Two scenarios as discussed in subsection 4.1.2: Ascending (AS) and Descending (DS) scenarios
- Six combinations of IMFs for each scenario as shown in Table 3
- Five time durations for the validation dataset: 2, 4, 6, 8, and 10 seconds

This is a total of $3*2*6*5 = 180$ combinations, but because ascending with IMFs 1 to 6 is the same as descending with IMFs 6 to 1 (the same IMFs just in a different order), there is a total of 165 experiments. Table 5 shows the number of features for each of the non-linear decomposition methods. For instance, in AS $1-5$, five IMFs are extracted for each original feature, IMFs 1 to 5; therefore, there are $5 * 21 = 105$ features.

Considering both linear and non-linear techniques, there is a total of 285 experiments. All components were implemented in Python. The preprocessing for all experiments was carried out on a computer with Ubuntu OS, AMD Ryzen 4.20 GHz processor, 128 GB DIMM RAM, and four NVIDIA GeForce RTX 2080 Ti 11GB graphics cards. CNNs were implemented with 'scikit-learn' library [56] and trained on Compute Canada (Cedar Cluster) resources. GPU acceleration was utilized to accelerate CNN training; nevertheless, the trained models do not require significant resources.

Table 4 Number of Features: SWT

LoD	Scenario F	Scenario R
1	42	21
3	84	63
5	126	105

Table 5 Number of Features: Non-linear Decomposition

# IMFs	AS (IMFs)	DS (IMFs)	# Features
1	1	6	21
2	1-2	6-5	42
3	1-3	6-4	63
4	1-4	6-3	84
5	1-5	6-2	105
6	1-6	6-1	126

4.6 Results

This section first presents accuracy, compares scenarios, analyzes the impact of personalization time on linear techniques, and then carries the same analysis for the non-linear ones. Next, linear and non-linear techniques are compared with statistical significance tests and the effect of activity type is examined. Finally, the proposed approach is compared with other machine learning approaches and the results are discussed.

4.6.1 Accuracy of Linear Decomposition Techniques

Tables 6 to 9 show accuracy for db1, db2, db3, and db4. Additionally, average and standard deviation (STD) for all models with the same personalization time are included. For example, the average of 85.77% for 2 s in Table 7 is the average of different models with 2 s personalization.

With db1, Table 6, the best accuracy of 88.8% was achieved with LoD 3, Scenario F, and 4 s validation set. Accuracy increases when LoD changes from 1 to 3 or from 1 to 5. For all experiments but two, the accuracy slightly decreases from Scenario F to Scenario R. In terms of the personalization time, there is no notable accuracy difference between durations.

Results for SWT with db2 are presented in Table 7: the best accuracy of 87.7% is achieved with LoD 5, Scenario F, and 6 s validation. Comparing Full (F) and Reduced (R) scenarios with the same LoD and time duration, it can be observed that the accuracy reduces in 10 out of 15 cases, but the difference is mostly small. Although the average accuracy 85.97% for 10 s is higher than for other durations, the standard deviation is higher indicating greater variability among experiments with 10 s validation time.

Table 8 shows the results for SWT with db3: the best value 87.4% is obtained with three different experiments as indicated with bold letters. As

Table 6 SWT: Accuracy for db1 Decomposition

LoD	Scenario	2 Secs	4 Secs	6 Secs	8 Secs	10 Secs
1	F	84.7	84.8	86	85	84.8
1	R	83.6	82.9	83.8	84.9	84.7
3	F	85.9	**88.8**	87.2	88.7	87.9
3	R	88.4	88.2	87.1	87.3	87.3
5	F	88.2	88.3	88.1	88.4	88.1
5	R	87.3	86.4	88.3	87.3	87.5
Avg.		**86.34**	**86.56**	**86.75**	**86.94**	**86.72**
STD		**1.8**	**2.14**	**1.51**	**1.5**	**1.4**

Table 7 SWT: Accuracy for db2 Decomposition

LoD	Scenario	2 Secs	4 Secs	6 Secs	8 Secs	10 Secs
1	F	85	84.2	84.4	85.6	85.6
1	R	84	85.3	83.9	84.3	82.5
3	F	86.1	86.8	86	85.5	87.4
3	R	86.4	86.4	87	86.1	86.7
5	F	86.9	84.6	**87.7**	87.4	87.2
5	R	86.1	85.7	86	86.8	86.4
Avg.		**85.77**	**85.5**	**85.83**	**85.95**	**85.97**
STD		**0.95**	**0.93**	**1.3**	**0.98**	**1.64**

Table 8 SWT: Accuracy for db3 Decomposition

LoD	Scenario	2 Secs	4 Secs	6 Secs	8 Secs	10 Secs
1	F	85.3	85.6	85.4	85.3	86
1	R	84.6	84.4	84	82.8	83.5
3	F	86.3	84.9	86.2	85.1	**87.4**
3	R	84.9	86.2	85.5	86.3	**87.4**
5	F	86.4	86.9	87.1	86.4	86.7
5	R	85.9	86.3	85.7	**87.4**	86.5
Avg.		**85.56**	**85.71**	**85.64**	**85.56**	**86.24**
STD		**0.67**	**0.87**	**0.94**	**1.45**	**1.34**

Table 9 SWT: Accuracy for db4 Decomposition

LoD	Scenario	2 Secs	4 Secs	6 Secs	8 Secs	10 Secs
1	F	85.1	86	85.7	85.6	84.7
1	R	82.3	84	82.5	83.1	84.2
3	F	85.8	87.6	87.3	87.3	86.7
3	R	84.7	85.7	85.9	84.7	86.4
5	F	86.6	87.2	87.1	86.6	87.3
5	R	87	84.6	86.3	**87.7**	85.7
Avg.		**85.24**	**85.83**	**85.79**	**85.86**	**86.24**
STD		**1.52**	**1.29**	**1.6**	**1.56**	**1.1**

LoD 3 scenario R has fewer features (63 features) than the other two (84 and 105 features), LoD 3 scenario can be considered better than the other two that achieved the same accuracy. The average accuracy for all validation set durations are similar, but shorter ones, 2-6 s have smaller standard deviation indicating more consistent results across experiments.

Finally, Table 9 displays the results for SWT with db4: the highest accuracy 87.7% is achieved with LoD 5, scenario R, and 8 s. The 10 s validation shows slightly better average accuracy than others while the standard deviation is high (over one) for all durations.

As seen from Tables 6 to 9, among linear techniques (Stationary Wavelet Transform), the best accuracy of 88.8% was achieved with db1, Scenario F, LoD 3, and 4 s personalization.

4.6.2 Comparison of Linear Decomposition Scenarios

This subsection compares linear decomposition scenarios F and R. Graphs in Fig. 6 show the accuracy for the two scenarios with respect to the four dbs and three LoDs: left column shows results for 2 s and the right column for 10 s validation set. The remaining durations are omitted as they follow similar patterns. Overall, scenario F performs better than scenario R. With LoD 1, scenario F outperforms scenario R for all dbs while for LoD 3 and LoD 5, scenario F has higher accuracy for most dbs.

Although for most dbs and validation set durations, scenario R has a lower accuracy than scenario F, the difference is small: the average differences between scenarios R and F for LoD 1, 3, and 5 are 1.48%, 0.32%, and 0.62%, respectively. Because the difference is so small, scenario R may be more desirable because of a fewer number of features, and thus, reduced training time.

4.6.3 Effect of Personalization Time on Linear Decomposition Techniques

Having a large validation set could potentially lead to improved personalization and thus, increased accuracy. On the other side, this dataset should be as small as possible to achieve ease of personalization.

Fig. 7 examines how the accuracy changes for each LoD and scenario as the duration of the personalization time increases. Only the graphs for db1 and db4 are shown as other dbs follow a similar pattern. From this figure, there are no easily observable and consistent patterns across all scenarios, dbs, and LoDs. If we compare validation time of 2 s with 8 s or with 10 s, most LoDs and scenarios exhibit an increase in accuracy, but not all. Overall, 8 and 10 s personalization time is achieving slightly better results than the other durations.

4.6.4 Accuracy of Non-linear Decomposition Techniques

Table 10 shows results for EMD with the linear spline while Table 11 presents the results of

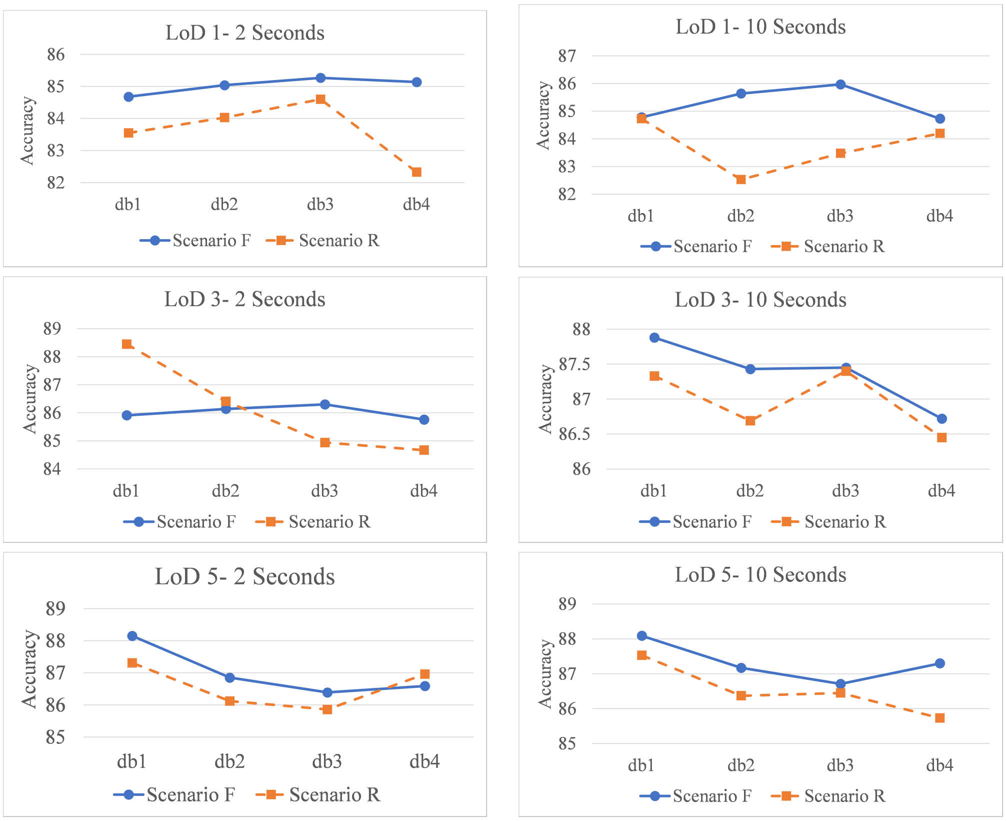


Fig. 6 SWT: Comparison of Scenarios F and R

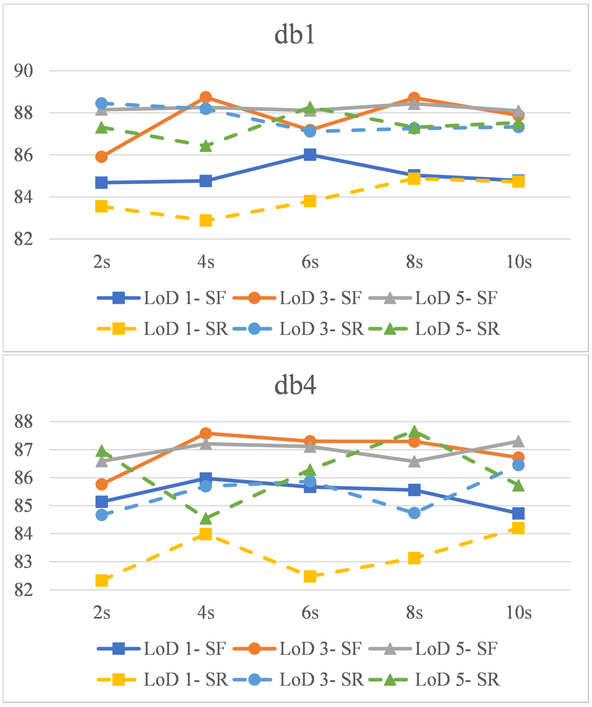


Fig. 7 Effect of Personalization Time on SWT Accuracy

EMD with the cubic spline. Average and STD accuracy obtained with different models with the same personalization time are also included.

For the linear spline, the best accuracy of 82.8% is achieved with all six IMFs and 8 s validation, irrelevant of the scenario. STD for descending scenario, for each validation duration, is higher than 18% while for ascending scenario, it is lower than 13%, which indicates that the ascending scenarios are achieving more consistent results. The best average accuracy of 72.13% is achieved with ascending scenario and 10 s validation.

With the cubic spline, as seen from Table 11, the highest accuracy of 91.2% is achieved with all IMFs and 6 s validation. For both, ascending and descending scenarios, accuracy increases as IMFs are added. STD for both scenarios is high, larger for the descending scenario, indicating high variability in obtained classifications. The average accuracy for descending scenarios is slightly lower than for ascending scenarios.

Table 12 presents the accuracy for EEMD methods including the average and STD for each

Table 10 EMD: Accuracy for Linear Spline

	IMFs	2 Secs	4 Secs	6 Secs	8 Secs	10 Secs
Ascending Scenario	1	45.9	46	44.5	45.1	45.4
	1-2	65.8	67	68.1	68.9	69.1
	1-3	69.2	70.9	75.2	68.4	78
	1-4	71.6	70.3	71.9	77.2	78.9
	1-5	75.3	76	78.7	76.9	79.6
	1-6	77.8	80.6	80	**82.8**	81.8
	Average	**67.6**	**68.46**	**69.73**	**69.88**	**72.13**
	STD	**10.45**	**10.95**	**11.97**	**12.15**	**12.6**
Descending Scenario	6	21	18.4	20.3	18.8	21
	6-5	50.6	47.2	46.5	45	46.7
	6-4	53.8	49	54.1	53.8	54.5
	6-3	61	60.8	60.5	57.4	65.9
	6-2	74.6	72.1	77.3	78.7	78.4
	6-1	77.8	80.6	80	**82.8**	81.8
	Average	**56.46**	**54.68**	**56.45**	**56.1**	**58.1**
	STD	**18.7**	**20.1**	**20.1**	**21.4**	**20.6**

Table 11 EMD: Accuracy for Cubic Spline

	IMFs	2 Secs	4 Secs	6 Secs	8 Secs	10 Secs
Ascending Scenario	1	45.7	45.4	45.9	46.1	44.8
	1-2	54.6	52.4	54.2	52.7	53.1
	1-3	60.1	58.3	61.8	59.5	59.4
	1-4	76.3	78.5	77.2	77.9	78.5
	1-5	84.2	84.7	86.3	87.7	86
	1-6	86.8	89.1	**91.2**	86.1	87.6
	Average	**67.95**	**68.07**	**69.43**	**68.33**	**68.23**
	STD	**15.4**	**16.74**	**16.6**	**16.3**	**16.6**
Descending Scenario	6	23.4	24.5	24.6	22.3	26
	6-5	52.6	48.9	46.9	47	50.8
	6-4	71	75.3	77.7	73.4	76.5
	6-3	82.1	82.8	81.3	84.2	83.3
	6-2	84.5	85.9	85.6	88.4	85.9
	6-1	86.8	89.1	**91.2**	86.1	87.6
	Average	**66.73**	**67.75**	**67.88**	**66.9**	**68.35**
	STD	**22.5**	**23.4**	**23.9**	**24.3**	**22.6**

personalization time. The highest accuracy of 63.8% is achieved with all six IMFs and a personalization time of 10 s. AS outperforms DS for the validation set of 2 and 4. In AS, the accuracy increases as the IMFs are added, while for DS, accuracy mostly increases with the addition of IMFs, but not always.

From Tables 10 to 12, it can be observed that among non-linear techniques (EMD and EEMD), the best accuracy of 91.2% was achieved with EMD with cubic spline, Ascending/Descending Scenario, and all 6 IMFs.

4.6.5 Comparison of Non-linear Scenarios

Fig. 8 compares ascending and descending scenarios for 2 and 10 s personalization time. Graphs for 4, 6, and 8 s exhibit similar patterns and thus are not included. Notice that the same number of IMFs is used for the ascending and descending scenarios; however, the included components are different. For example, for three IMFs, AS includes IMFs 1, 2, and 3 while DS includes IMFs 6, 5, and 4 as shown in Table 3. As seen from Fig. 8, for all decomposition techniques and all scenarios, the accuracy raises when the number of IMFs increases. For both personalization durations, when only one IMF is used, ascending

Table 12 Accuracy for EEMD

	IMFs	2 Secs	4 Secs	6 Secs	8 Secs	10 Secs
Ascending Scenario	1	28.2	25.9	27.9	28.3	29.2
	1-2	37.2	35.6	33.9	34	33.6
	1-3	37.2	37.5	38.6	38.3	37.9
	1-4	46.6	51.5	44.1	46.9	48.2
	1-5	51.1	53.3	52.2	53.7	54.6
	1-6	54	54.1	63	58.4	**63.8**
	Average	**42.38**	**42.98**	**43.28**	**43.27**	**44.55**
	STD	**8.98**	**10.63**	**11.65**	**10.68**	**12.14**
Descending Scenario	6	19.1	19	16.2	19.6	22
	6-5	26.3	28.2	24.5	28.3	28.2
	6-4	38.3	39.8	43.2	42.6	41.1
	6-3	50.2	58.2	57	58.4	50.2
	6-2	54.7	53.2	56.6	56.6	51.7
	6-1	54	54.1	63	58.4	**63.8**
	Average	**40.43**	**42.08**	**43.41**	**43.98**	**42.83**
	STD	**13.8**	**14.48**	**17.5**	**15.4**	**14.3**

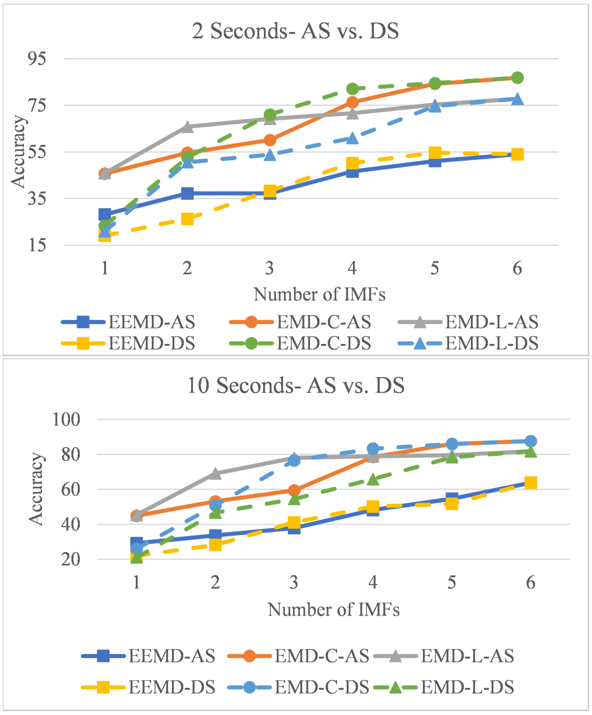


Fig. 8 Non-linear techniques: Ascending and Descending Scenarios with 2 and 10 s personalization

scenarios (solid lines) are better than descending (dashed lines). When more IMFs are added, DS is catching up with AS, and is even slightly better for some numbers of IMFs. The same Fig. 8 compares EMD with two different splines, namely linear (EMD-L) and cubic (EMD-C). With a fewer number of IMFs, linear spline achieves better accuracy, but when IMFs are added, the cubic spline performs better. Overall, out of all non-linear techniques, EMD with cubic spline and six IMFs achieves the highest accuracy.

4.6.6 Effect of Personalization Time on Non-Linear Decomposition Techniques

Fig. 9 examines the effect of personalization time on Cubic EMD technique. Only the graphs for Cubic EMD with Ascending and Descending scenarios are included as other techniques, EMD with linear spline and EEMD, follow a similar pattern.

The accuracy remains quite stable with the increase in personalization time demonstrating that the proposed personalization can work successfully even with as little as 2 s of the target subject data per activity. In both, ascending and descending scenarios, accuracy is higher when more IMFs are included, with six IMFs achieving the best accuracy for almost all personalization time durations.

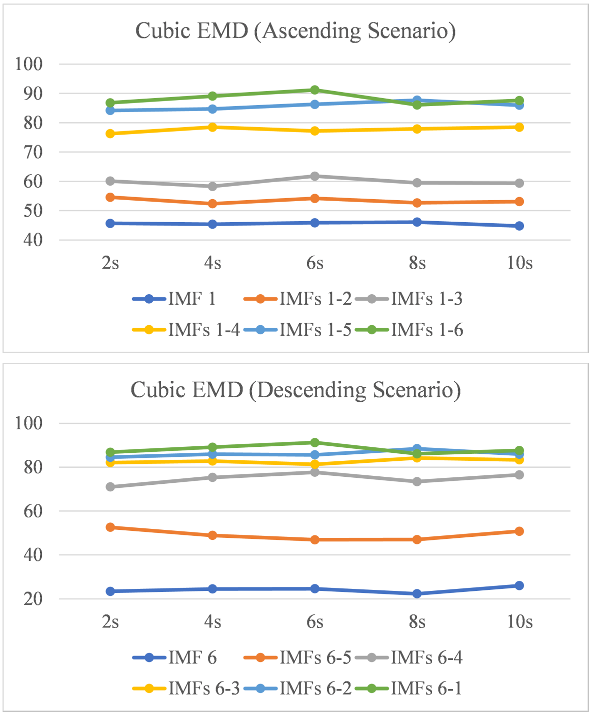


Fig. 9 Effect of Personalization Time on Cubic EMD Technique

4.6.7 Comparing the Best Linear and Non-linear Techniques: Statistical Significance Tests

In this subsection, we examine if the difference between the best linear and the best non-linear technique is significant. As seen from Tables 6 to 9, the highest accuracy of 88.8% among all linear preprocessing techniques was achieved by SWT with db1, LoD 3, Scenario F, and 4 s personalization. From non-linear techniques (Tables 10 to 12), the best accuracy of 91.2% was obtained by EMD with cubic spline, six IMFs, and 6 s personalization. Because the accuracy of the CNNs can vary slightly depending on factors such as the initial weights and the order of samples during training, we repeat the process of training and testing the CNN 100 times for each of the two best models, linear and non-linear. Finally, we carry out the statistical tests to determine if there is a significant difference between the two.

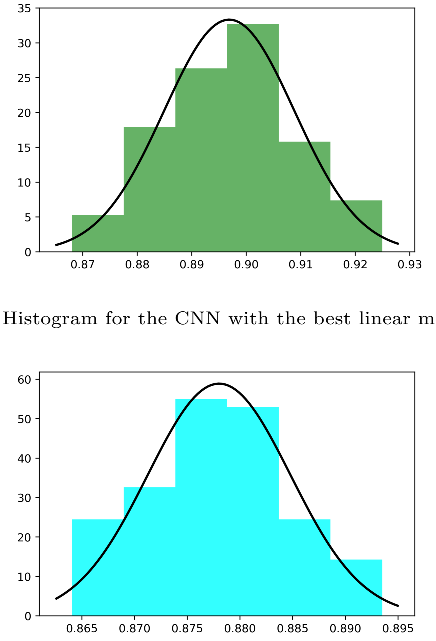
To determine which statistical test to use, histograms of the accuracy are plotted first as shown in Fig. 10. The distribution from those histograms appears to follow the normal distribution, but to confirm this, Shapiro-Wilk test is applied. The criteria for accepting/rejecting the null hypothesis "Samples are not from the normal distribution" is as follows:

$$\begin{cases} p <= \alpha & \text{Reject H0; Distribution is Not Normal} \\ p > \alpha & \text{Fail to Reject H0; Distribution is Normal} \end{cases} \tag{9}$$

where p is the statistic value and α is the significance level.

The results of the Shapiro-Wilk test are shown in Table 13. Because p-value for both, linear and non-linear decomposition, is greater than $\alpha = 0.05$, we conclude that they follow Gaussian distribution.

As the results of linear and non-linear decomposition follow a normal distribution, a t-test can be used to determine if there is a significant difference between the two techniques. The null



(a) Histogram for the CNN with the best linear model

(b) Histogram for the CNN with the best non-linear model

Fig. 10 Accuracy Histograms

Table 13 Results of Shapiro-Wilk test

	Linear + CNNs	Non-Lienar + CNNs
Statistics	0.987	0.995
P-value	0.439	0.971
Result	Gaussian Distribution (fail to reject H0)	Gaussian Distribution (fail to reject H0)

hypothesis is now:

$$\text{H0: Ave (non-linear+CNN)} = \text{ Ave (linear+CNN)} \tag{10}$$

For the t-test, statistics value is 3.65 corresponding to $p - value = 2.48e - 30$ at significance level 0.05. Because the p-value is lower than 0.05, we can reject the null hypothesis, which means that the linear and non-linear techniques do not have the same mean. Finally, we can infer that CNN with non-linear decomposition is statistically better than CNN with linear decomposition.

4.6.8 Effect of Activity Type

Table 14 shows the combined confusion matrix for all 10 people and the best model, EMD with 6 IMFs and 6 seconds for personalization. Each field in the matrix is obtained by adding the numbers from the corresponding fields in each person confusion matrix. It can be observed that for some activities the misclassification is much higher than for others. The most frequently confused activities are jogging (activity 10) and running (activity 11) which is understandable as there is no firm boundary between those two activities: 6069 times jogging was detected as running and 2159 times running was classified as jogging. Nevertheless, the model was highly successful in detecting the majority of the activities.

4.7 Comparison with other machine learning techniques

This subsection compares the proposed approach with other ML-based HAR approaches. Two algorithms have been considered: K-Nearest Neighbour (KNN) and SVM. Note that the work of [33] has already shown that CNN outperforms FFNN in HAR, in particular on the MHEALTH dataset. With each of the two considered algorithms, two data preprocessing strategies have been employed. In the first strategy, referred here as *raw data*, the sliding window technique is applied and the raw data is used for the classification. In the second strategy, referred to as *extracted features*, the sliding window technique is applied, and then the features are extracted from each individual data window as in the work of Khatun and Morshed [57]. For each original feature obtained from sensors, nine features are extracted from created windows: mean, min, max, standard deviation, median, skewness, energy, kurtosis, and interquartile range [57]. These features with corresponding labels are then used for model training.

All experiments have been evaluated with the LOSOCV technique. Table 15 shows the results of KNN with the raw data strategy (CPU implementation). KNN does not have a separate training stage; thus, the table shows activity detection time for all folds of the LOSOCV. As seen from the table, when the number of considered nearest neighbors K increases, the processing time increases with minor changes in the activity recognition accuracy. However, when the window size increases, so does the accuracy: the best accuracy was achieved for window size 50 and $K = 3$.

Table 16 depicts results for KNN with extracted features. The accuracy of this approach

Table 14 Combined confusion matrix for all ten people (MHEALTH Dataset, EMD IMF1-6 and 6 s personalization)

		Prediction											
	Activity	1	2	3	4	5	6	7	8	9	10	11	12
Real Classes	1	26220	869	0	0	0	0	631	0	0	0	0	0
	2	49	25564	593	566	0	364	519	0	0	0	65	0
	3	103	0	26816	7	0	0	745	0	0	0	0	49
	4	**1191**	27	0	25682	113	70	132	75	0	107	323	0
	5	**2068**	725	34	**1455**	22467	31	0	248	0	108	19	565
	6	60	14	0	11	0	24909	295	19	7	0	0	0
	7	0	**1155**	672	19	0	111	24325	19	2	138	0	0
	8	0	0	0	0	0	**1931**	0	24080	67	254	0	5
	9	11	0	0	0	1	0	116	67	27462	25	0	38
	10	0	0	0	0	0	0	319	133	0	20972	**6069**	227
	11	0	0	0	0	0	0	67	55	1	**2159**	25302	136
	12	0	0	0	20	463	83	0	6	17	163	241	6349

1: Standing still, 2: Sitting and relaxing, 3: Lying down, 4: Walking, 5: Climbing stairs, 6: Waist bends forward, 7: Frontal elevation of arms, 8: Knees bending (crouching), 9: Cycling, 10: Jogging, 11: Running, 12: Jump front & back

is lower than when KNN is used with sliding windows and raw data (Table 15). As the size of the sliding window increases, so does the accuracy, while the impact of the considered nearest neighbors K is minimal. Increasing the window size results in a decreased number of samples (as there is no window overlap) as well as in reduced detection time.

For the SVM classifier, training with the sliding windows and raw data was very slow, with a training time of over 20 hours even for a single fold of LOSCOCV; therefore, this was considered an unfeasible approach, and further experiments were not carried out. For extracted features, training SVM was faster but the overall accuracy was below 60%; therefore further analysis is not presented here.

Table 15 KNN with Sliding Windows and Raw Data

Window Size (Other Details)	K	Accuracy	Time
5 (Samples: 342715 Features: 105)	3	81.6	1h, 40m, 20s
	5	81.8	1h, 42m, 11s
	7	81.8	1h, 44m, 36s
	9	81.8	1h, 45m, 58s
10 (Samples: 342115 Features: 210)	3	83.5	4h, 10m, 13s
	5	83.5	4h, 16m, 53s
	7	83.6	4h, 20m, 38s
	9	83.6	4h, 28m, 52s
50 (Samples: 337315 Features: 1050)	3	89.4	24h, 40m, 2s
	5	89.3	25h, 26m, 10s
	7	89.3	26h, 14m, 46s
	9	89.2	27h, 25m, 40s

To compare these results with our approach, Table 17 shows time obtained with our best personalized model: CNN with Cubic EMD with 6 IMFs, and 6 s for personalization. Time for both, CPU and GPU-based implementations are shown separately for training and testing. Presented time values include all ten folds (subjects) of LOSOCV and all samples. Consequently, with our approach, the activity cumulative detection time for all 307,195 samples is $29s$ on CPU and $23s$ on GPU. Thus, for a single sample, the detection time is under 0.1 ms.

The accuracy of KNN with sliding window and raw data (Table 15) reaches the accuracy of our approach (Table 11) but KNN does not have a separate training: to achieve comparable accuracy to our approach, KNN needs a window size of 50 what results in over 24 hours for activity detection for all test samples. As our approach carries out activity recognition much faster, it is an overall better approach.

5 Discussion

The presented experiments demonstrate that CNN with signal processing techniques and the proposed personalization approach achieves high HAR accuracy. In both, linear and non-linear decomposition techniques, the achieved accuracy depends on the number of the created features

Table 16 KNN with Features Extracted from Sliding Windows

Window Size (Other Details)	K	Accuracy	Time
5 (Samples: 68588 Features: 189)	3	68	56s
	5	68.6	59s
	7	68.9	1m, 1s
	9	69	1m, 3s
10 (Samples: 34284 Features: 189)	3	69.5	14s
	5	70.2	15s
	7	70.2	16s
	9	70.2	17s
50 (Samples: 6812 Features: 189)	3	83.3	1s
	5	83.3	1s
	7	83.3	1s
	9	83.4	1s

Table 17 Personalized model: CNNs with 6 IMFs, Cubic EMD, and 6 s personalization time

CNN+Personalization	Time (train)	Time (Test)
CPU Samples: 307195 Features: 126	4h, 30m, 45s	29s
GPU Samples: 307195 Features: 126	3h, 27m, 16s	23s

(levels of decomposition or IMFs) for CNN training.

For linear techniques, as observed from Fig. 7 and Tables 6 to 9, increasing the level of decomposition for almost all personalization time durations improves activity recognition. In most cases, Scenario F outperforms Scenario R; however, Scenario R achieves very similar accuracy to Scenario F but with a fewer number of features, and therefore reduced training time.

Similarly, as seen in Fig. 8, the recognition accuracy for non-linear techniques increases with the number of features, in this case, the number of IMFs. Adding features has a larger impact on non-linear techniques than on linear techniques. For example, EMD with cubic spline and ascending scenario has accuracy below 50% with one IMF, which increases to over 86% for six IMFs (Fig. 8). For linear scenarios, the accuracy increase from LoD 1 to LoD 5 is much smaller: in the range of 82% to 88%. For a low number of IMFs, the ascending scenario is better than descending scenario (Fig. 8) but this pattern revertses when the number of IMFs increases to five. With six IMFs, the two scenarios become the same.

Finally, the proposed approach is compared with the state-of-the-art CNN-based HAR technique [33]. Mentioned study [33] examined different CNN and FFNN architectures with various pre-processing techniques including the sliding window and vector magnitude. In contrast, our study employs the signal processing techniques with CNNs and proposes a novel personalization approach. Here we compare our approach with the best architecture from that study [33]: CNN (two convolutional, two pooling, and two fully-connected layers) with sliding window and vector magnitude pre-processing. This architecture is compared to the best linear model SWT-db1-3LoD and the best non-linear model EMD-C-AS/DS in Table 18.

The best accuracy of CNN with the sliding window, vector magnitude, and without personalization [33] is 85.1% whereas the best accuracy achieved in our study is 91.2%. Even without personalization, the best non-linear technique EMD-C-AS/DS obtains better results than CNN with the sliding window and vector magnitude. Note that the scenario here is denoted as AS/DS because with six IMFs, DS and AS scenarios are the same.

This Table 18 also examines adding our personalization technique to the CNN with sliding window and vector magnitude [33]: personalization increases the accuracy to around $87 - 88\%$ depending on the personalization time. Comparing the impact of personalization on each of the considered techniques, CNN with the sliding window and vector magnitude, linear (SWT-db1-3LoD), and non-linear (EMD-C-AS/DS) decomposition models, we can observe that adding personalization increase the accuracy for each of the three models. With the non-linear approach, even without personalization, the accuracy is already high, which shows that this approach is capable of extracting more valuable features than the other two.

Table 18 shows that the proposed feature extraction approach achieves better accuracy than the sliding window technique with vector magnitude. Another point to highlight is that with window size 50 and vector magnitude, the number of features is 350 [33] while for SWT-db1-3 LoDs (Scenario F) and EMD-Cubic-AS/DS (IMFs 1-6), the number of features is 84 and 126, respectively. Consequently, the proposed approach is not only

Table 18 Comparison of the proposed technique with the state-of-the-art CNN-based approach

	Without	With Personalization				
	Pers.	2s	4s	6s	8s	10s
CNN with siding window and vector magnitude [33]	85.1	88.4	88	87.8	87.7	88
SWT-db1-3 LoDs (Scenario F)	84.96	85.9	88.8	87.2	88.7	87.9
EMD-Cubic-AS/DS (IMFs 1-6)	86.98	86.8	89.1	**91.2**	86.1	87.6

Table 19 Comparison of the proposed technique with the state-of-the-art CNN-based approach for WISDM dataset

	Without	With Personalization				
	Pers.	2s	4s	6s	8s	10s
CNN with siding window and vector magnitude [33]	50.17	–	52.57	52.59	53.63	53.9
EMD-Cubic-AS/DS (IMFs 1-6)	52.14	53.43	53.47	57.6	55.2	54.84

able to achieve higher accuracy, but it also reduces CNN training time due to the reduced number of features.

Note that all results presented in this study are obtained with LOSOCV evaluation and are not comparable with values reported in literature if a different evaluation technique is used. It has been shown that differences between the evaluation techniques can exceed 14% [33]. CNN with sliding window and vector magnitude achieved 85.1% accuracy when evaluated with LOSOCV, and 99.85% when evaluated with traditional 10-fold cross-validation [33].

To examine the behavior of the proposed approach on a larger activity-based dataset, we have used WISDM dataset [58]. This dataset was gathered from 51 volunteers as they executed 18 different activities for three minutes each. Data from sensors is recorded for two devices, phone and watch, and two sensor types, accelerometer, and gyroscope, for each device. All sensor modalities operated at a sampling rate of 20 Hz. Table 19 shows the results for WISDM dataset; accuracy of our approach is compared to the CNN with sliding window and vector magnitude [33] with and without the personalization Note that the personalization approach with 2 seconds is not applicable for the CNN with sliding window and vector magnitude (window size: 50) because the size of the validation dataset becomes 40 (2 sec with 20 Hz sampling = 40 readings) which is less than the window size.

As seen from Table 19, for the CNN with sliding window and vector magnitude [33], the accuracy increases as the personalization time increases. For our best model, EMD with 6 IMFs, the personalization produces better accuracy than the model without the personalization, with the best results achieved for 6 s. Nevertheless, the personalization improves the results irrelevant of the base model, and moreover, our signal-based solution achieves better accuracy than the state-of-the-art CNN [33]. The accuracy for WISDM is much lower than for MHEALTH dataset which is to be expected due to the complexity and diversity of the activities in WISDM dataset. The results reported here cannot be compared directly with other studies as they use a less strict evaluation technique [59] or they only consider some of the recorded activities [60, 61].

For all experiments, we have used the same CNN architecture as shown in Fig. 3 because that was the best CNN architecture obtained in the previous study [33]. Moreover, the same hyperparameters have been used as in the mentioned study [33]. Even without tuning this architecture, in this paper we presented results from 285 models for MHEALTH dataset, which with 10 subjects equals 2850 experiments. A large number of alternatives makes comparing individually tuned architectures difficult while further tuning could result in increased accuracy of some variants presented.

Note that data for HAR is collected in controlled experiments, therefore, it leads to a balanced dataset. In the case of imbalanced classes, what is the expected scenario in real-world applications, using F1-measure instead of accuracy would be a better way of selecting the best model; however, this situation could also be remedied with under-sampling and over-sampling techniques.

6 Conclusion

This paper proposes a personalized approach for human activity recognition from wearable sensors. The multi-modal sensor data from the magnetometer, accelerometer, and gyroscope are first processed by signal processing techniques to extract features for machine learning. Next, the convolutional neural network fuses the extracted features to obtain the final activity classification. A few seconds of the target subject data are used to personalize the model by selecting the model version best suited to that subject.

Experiments examined linear techniques including SWT variants with mother wavelets db1, db2, db3, and db4 as well as non-linear techniques such as EMD with a linear and cubic spline. For linear techniques, one, three, and five levels of decomposition were considered while for non-linear techniques, different numbers of IMFs were examined. Also, the paper examines how much data from the target subject is required for personalization.

The results show that EMD with cubic spline, six IMFs, and 6 s personalization archives the best accuracy of 91.2%. In comparison, the state-of-the-art CNN-based technique achieves accuracy around 85.1% [33]. Mostly, for all signal decomposition techniques, adding more features increases accuracy but those added features also increase computational time for both, feature extraction as well as CNN training. On the other hand, personalization time does not have as strong an impact on accuracy, and increasing personalization time does not always result in increased accuracy. Finally, experiments demonstrated that the proposed personalization technique can improve the accuracy of existing CNN-based approaches even when used without the signal processing component.

Future work will evaluate the proposed approach on different datasets and with a larger number of subjects. Moreover, a hybrid of time- and frequency-domain features will be examined.

Acknowledgements: This work was supported by Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-2018-06222.

Data Availability: MHEALTH [54] dataset is available from http://archive.ics.uci.edu/ml/datasets/mhealth+dataset.

WISDM [58] dataset is available from: https://archive.ics.uci.edu/ml/datasets/WISDM+Smartphone+and+Smartwatch+Activity+and+Biometrics+Dataset+.

Declarations

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

References

[1] Giuliano Grossi, Raffaella Lanzarotti, Paolo Napoletano, Nicoletta Noceti, and Francesca Odone. Positive technology for elderly well-being: A review. *Pattern Recognition Letters*, 137:61–70, 2020.

[2] Qimeng Li, Raffaele Gravina, Ye Li, Saeed H Alsamhi, Fangmin Sun, and Giancarlo Fortino. Multi-user activity recognition: Challenges and opportunities. *Information Fusion*, 63:121–135, 2020.

[3] Henry Friday Nweke, Ying Wah Teh, Ghulam Mujtaba, and Mohammed Ali Al-Garadi. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Information Fusion*, 46:147–170, 2019.

[4] Saeed Mohsen, Ahmed Elkaseer, and Steffen G Scholz. Human activity recognition using k-nearest neighbor machine learning algorithm. In *Proceedings of the International Conference on Sustainable Design and Manufacturing*, pages 304–313. Springer, 2021.

[5] Nadeem Ahmed, Jahir Ibna Rafiq, and Md Rashedul Islam. Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors*, 20(1), 2020.

[6] Chunyu Hu, Yiqiang Chen, Lisha Hu, and Xiaohui Peng. A novel random forests based class incremental learning method for activity recognition. *Pattern Recognition*, 78:277–290, 2018.

[7] Parviz Asghari, Elnaz Soleimani, and Ehsan Nazerfard. Online human activity recognition employing hierarchical hidden markov models. *Journal of Ambient Intelligence and Humanized Computing*, 11(3):1141–1152, 2020.

[8] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40, 2021.

[9] Juan M Gandarias, Alfonso J Garcia-Cerezo, and Jesus M Gomez-de Gabriel. CNN-based methods for object recognition with high-resolution tactile sensors. *IEEE Sensors Journal*, 19(16):6872–6882, 2019.

[10] Erdenebileg Batbaatar, Meijing Li, and Keun Ho Ryu. Semantic-emotion neural network for emotion recognition from text. *IEEE Access*, 7:111866–111878, 2019.

[11] Muhammad Zubair Asghar, Ammara Habib, Anam Habib, Adil Khan, Rehman Ali, and Asad Khattak. Exploring deep neural networks for rumor detection. *Journal of Ambient Intelligence and Humanized Computing*, 12(4):4315–4333, 2021.

[12] Khushwant Rai, Farnam Hojatpanah, Firouz Badrkhani Ajaei, and Katarina Grolinger. Deep learning for high-impedance fault detection: Convolutional autoencoders. *Energies*, 14(12), 2021.

[13] Mohammad Mehedi Hassan, Md Golam Rabiul Alam, Md Zia Uddin, Shamsul Huda, Ahmad Almogren, and Giancarlo Fortino. Human emotion recognition using deep belief network architecture. *Information Fusion*, 51:10–18, 2019.

[14] Yongming Huang, Kexin Tian, Ao Wu, and Guobao Zhang. Feature fusion methods research based on deep belief networks for speech emotion recognition under noise condition. *Journal of Ambient Intelligence and Humanized Computing*, 10(5):1787–1798, 2019.

[15] Ananda Mohon Ghosh and Katarina Grolinger. Edge-cloud computing for internet of things data analytics: Embedding intelligence in the edge with deep learning. *IEEE Transactions on Industrial Informatics*, 17(3):2191–2200, 2020.

[16] Martin Gjoreski, Vito Janko, Gašper Slapničar, Miha Mlakar, Nina Reščič, Jani Bizjak, Vid Drobnič, Matej Marinko, Nejc Mlakar, Mitja Luštrek, et al. Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors. *Information Fusion*, 62:47–62, 2020.

[17] Zhen Qin, Yibo Zhang, Shuyu Meng, Zhiguang Qin, and Kim-Kwang Raymond Choo. Imaging and fusing time series for wearable sensor-based human activity recognition. *Information Fusion*, 53:80–87, 2020.

[18] Yasser Mohammad, Kazunori Matsumoto, and Keiichiro Hoashi. Primitive activity recognition from short sequences of sensory data. *Applied Intelligence*, 48(10):3748–3761, 2018.

[19] Sreenivasan Ramasamy Ramamurthy and Nirmalya Roy. Recent trends in machine learning for human activity recognition—a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), 2018.

[20] Hong He, Yonghong Tan, and Wuxiong Zhang. A wavelet tensor fuzzy clustering scheme for multi-sensor human activity recognition. *Engineering Applications of Artificial Intelligence*, 70:109–122, 2018.

[21] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, 108, 2020.

[22] BA Mohammed Hashim and R Amutha. Human activity recognition based on smartphone using fast feature dimensionality reduction technique. *Journal of Ambient Intelligence and Humanized Computing*, 12:2365–2374, 2021.

[23] Raffaele Gravina, Parastoo Alinia, Hassan Ghasemzadeh, and Giancarlo Fortino. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion*, 35:68–80, 2017.

[24] Shida Chen, Pengcheng Liu, Dazhen Tang, Shu Tao, and Taiyuan Zhang. Identification of thin-layer coal texture using geophysical logging data: Investigation by wavelet transform and linear discrimination analysis. *International Journal of Coal Geology*, 239, 2021.

[25] Dengsheng Zhang. Wavelet transform. In *Fundamentals of Image Data Mining*, pages 35–44. Springer, 2019.

[26] Sapna Pandit and Seema Sharma. On the use of wavelets for analysis of nanofluid flow and thermal transmission through asymmetric porous channel. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, pages 1–13, 2022.

[27] Diego HS Nolasco, Flavio B Costa, Eduardo S Palmeira, Denis K Alves, Benjamín RC Bedregal, Thiago OA Rocha, Ricardo LA Ribeiro, and Juliano CL Silva. Wavelet-fuzzy power quality diagnosis system with inference method based on overlap functions: Case

study in an ac microgrid. *Engineering Applications of Artificial Intelligence*, 85:284–294, 2019.

[28] Mubarak G Abdu-Aguye and Walid Gomaa. Competitive feature extraction for activity recognition based on wavelet transforms and adaptive pooling. In *IEEE International Joint Conference on Neural Networks*, 2019.

[29] Zhaohua Wu and Norden E Huang. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in adaptive data analysis*, 1(01):1–41, 2009.

[30] Dustin P Fairchild and Ram M Narayanan. Classification of human motions using empirical mode decomposition of human micro-doppler signatures. *IET Radar, Sonar & Navigation*, 8(5):425–434, 2014.

[31] Yiming Tian, Jie Zhang, Jie Wang, Yanli Geng, and Xitai Wang. Robust human activity recognition using single accelerometer via wavelet energy spectrum features and ensemble feature selection. *Systems Science & Control Engineering*, 8(1):83–96, 2020.

[32] Huile Xu, Jinyi Liu, Haibo Hu, and Yi Zhang. Wearable sensor-based human activity recognition method with multi-features extracted from hilbert-huang transform. *Sensors*, 16(12), 2016.

[33] Davoud Gholamiangonabadi, Nikita Kiselov, and Katarina Grolinger. Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. *IEEE Access*, 8:133982–133994, 2020.

[34] Ingrid Daubechies. *Ten lectures on wavelets*. Society for industrial and applied mathematics., 1992.

[35] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[36] Prakash K Ray, BK Panigrahi, PK Rout, Asit Mohanty, and Harishchandra Dubey. Detection of faults in power system using wavelet transform and independent component analysis. In *Computer, Communication and Electrical Technology: Proceedings of the International Conference on Advancement of Computer Communication and Electrical Technology*, 2017.

[37] I Omerhodzic, S Avdakovic, A Nuhanovic, and K Dizdarevic. Energy distribution of eeg signals: Eeg signal wavelet-neural network classifier.

[38] Dagimawi Eneyew, Miriam AM Capretz, Girma Bitsuamlak, and London Hydro. Predicting residential energy consumption using wavelet decomposition with deep neural network. 2020.

[39] Mingju Gong, Jin Wang, Yin Bai, Bo Li, and Lei Zhang. Heat load prediction of residential buildings based on discrete wavelet transform and tree-based ensemble learning. *Journal of Building Engineering*, 32, 2020.

[40] H Azzaoui, I Mansouri, and B Elkihel. Methylcyclohexane continuous distillation column fault detection using stationary wavelet transform & fuzzy c-means. *Materials Today: Proceedings*, 13, 2019.

[41] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, 454(1971):903–995, 1998.

[42] Yao Liu, Hongbin Pu, and Da-Wen Sun. Efficient extraction of deep image features using convolutional neural network (cnn) for applications in detecting and analysing complex food matrices. *Trends in Food Science & Technology*, 113:193–204, 2021.

[43] Munoz-Organero Mario. Human activity recognition based on single sensor square hv acceleration images and convolutional neural networks. *IEEE Sensors Journal*, 19(4):1487–1498, 2018.

[44] Seo Yul Kim, Hong Gul Han, Jin Woo Kim, Sanghoon Lee, and Tae Wook Kim. A hand gesture recognition sensor using reflected impulses. *IEEE Sensors Journal*, 17(10):2975–2976, 2017.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[46] Sojeong Ha, Jeong-Min Yun, and Seungjin Choi. Multi-modal convolutional neural networks for activity recognition. In *IEEE International conference on systems, man, and cybernetics*, 2015.

[47] Md Zia Uddin and Mohammad Mehedi Hassan. Activity recognition for cognitive assistance using body sensors data and deep

convolutional neural network. *IEEE Sensors Journal*, 19(19):8413–8419, 2018.

[48] Yuya Hanai, Jun Nishimura, and Tadahiro Kuroda. Haar-like filtering for human activity recognition using 3d accelerometer. In *IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, 2009.

[49] Zhenyu He and Lianwen Jin. Activity recognition from acceleration data based on discrete consine transform and SVM. In *IEEE International Conference on Systems, Man and Cybernetics*, 2009.

[50] Roland Assam and Thomas Seidl. Activity recognition from sensors using dyadic wavelets and hidden markov model. In *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2014.

[51] Anna Ferrari, Daniela Micucci, Marco Mobilio, and Paolo Napoletano. On the personalization of classification models for human activity recognition. *IEEE Access*, 8:32066–32079, 2020.

[52] Trang Thuy Vu and Kaori Fujinami. Understanding compatibility-based classifier personalization in activity recognition. In *Joint 8th IEEE International Conference on Informatics, Electronics & Vision and 3rd International Conference on Imaging, Vision & Pattern Recognition*, 2019.

[53] Ching-Yi Lin and Radu Marculescu. Model personalization for human activity recognition. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, 2020.

[54] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mHealthDroid: a novel framework for agile development of mobile health applications. In *International workshop on ambient assisted living*, 2014.

[55] Oresti Banos, Claudia Villalonga, Rafael Garcia, Alejandro Saez, Miguel Damas, Juan A Holgado-Terriza, Sungyong Lee, Hector Pomares, and Ignacio Rojas. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical engineering online*, 14(2):1–20, 2015.

[56] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[57] Saleha Khatun and Bashir I Morshed. Fully-automated human activity recognition with transition awareness from wearable sensor data for mhealth. In *2018 IEEE International Conference on Electro/Information Technology*, pages 0934–0938. IEEE, 2018.

[58] Gary M Weiss, Kenichi Yoneda, and Thaier Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.

[59] Seyed Ali Rokni, Marjan Nourollahi, and Hassan Ghasemzadeh. Personalized human activity recognition using convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[60] Hendrio Bragança, Juan G Colonna, Wesllen Sousa Lima, and Eduardo Souto. A smartphone lightweight method for human activity recognition based on information theory. *Sensors*, 20(7), 2020.

[61] Artur Jordao, Antonio C Nazare Jr, Jessica Sena, and William Robson Schwartz. Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. 2019.