Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05638-4

ORIGINAL ARTICLE

Feature learning using convolutional denoising autoencoder for activity recognition

Mohd Halim Mohd Noor[1]

Received: 14 January 2020 / Accepted: 16 December 2020

Abstract

Wearable technology offers a prospective solution to the increasing demand for activity monitoring in pervasive healthcare. Feature extraction and selection are crucial steps in activity recognition since it determines the accuracy of activity classification. However, existing feature extraction and selection methods involve manual feature engineering, which is time-consuming, laborious and prone to error. Therefore, this paper proposes an unsupervised feature learning method that automatically extracts and selects the features without human intervention. Specifically, the proposed method jointly trains a convolutional denoising autoencoder with a convolutional neural network to learn the underlying features and produces a compact feature representation of the data. This allows not only more accurate and discriminative features to be extracted but also reduces the computational cost and improves generalization of the classification models. The proposed method was evaluated and compared with deep learning convolutional neural networks on a public dataset. Results have shown that the proposed method can learn a salient feature representation and subsequently recognize the activities with an accuracy of 0.934 and perform comparably well to the convolutional neural networks.

Keywords Activity recognition · Denoising autoencoder · Deep learning · Feature learning

1 Introduction

Aging is commonly associated with loss of independence and impairment in physical functioning which is a major indicator of frailty. Physical activity plays a major role in healthy aging and is essential to maintain healthy functioning in elderly people [1]. In particular, physical activity can reduce the negative impact of frailty and reduce the risk of frailty in older adults [2]. It is also shown that physical activity reduces the risk of having chronic diseases such as cardiovascular disease, diabetes and osteoporosis [3]. In fact, a study has suggested that the efficacy of physical activity intervention is comparable to drug intervention in preventing and treatment of chronic diseases [4].

Wearable sensor-based activity recognition is a system that classifies gait data into human activities. Human activities can be grouped into basic activity and activity of daily living (ADL). Basic activity refers to basic body movements such as walking, standing and postural transition. ADLs are complex activities that typically involve objects such as eating, cooking and bathing. Activity recognition systems utilize wearable sensors such as accelerometers, gyroscopes and wireless communication to gather users' behavioral information which can be used to keep track of the level of activities being performed by users and estimate their energy expenditure. The wearable sensors can be attached either directly to the human body or indirectly such as embedding into clothes or wristwatches. The sensors generate signals when the user performs activities where the signals' characteristics describe the user's movement.

Activity recognition involves the use of artificial intelligence techniques to effectively recognize a range of different activities. Typically, the recognition process operates in two steps. First, features are extracted from segments of sensor data from a specific time window. Then, the features are used as inputs to machine learning classifiers to classify the window according to its

✉ Mohd Halim Mohd Noor
halimnoor@usm.my

[1] School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia

Published online: 01 January 2021

corresponding activity. The performance of activity recognition is heavily dependent on the choice of features that are used to model the classifiers. In previous studies, numerous types of features have been used for activity recognition with some researchers derive features directly from the time series signal by computing heuristic and statistical measurements and others derive frequency-domain features from Fourier transformed signal [5]. Heuristic features refer to features that are derived from the intuitive understanding of how a signal pattern is produced by an activity. For example, a walking signal exhibits a quasiperiodic pattern while sitting posture is characterized by a stationary signal. Time-domain features are derived by means of statistical analysis such as mean, variance and kurtosis. Both heuristic and time-domain features are lightweight and simple to compute, and studies have shown they can distinguish static postures and dynamic activities with high accuracy.

However, the hand-crafted features that are extracted and selected are heuristic and rely on expert knowledge of the domain. The features may be effective in certain specific settings, but the same features might fail to discriminate the activities in a more general environment. Furthermore, hand-crafted feature extraction and selection are time-consuming, laborious and prone to error and might still achieve suboptimal recognition performance. The main objective of this work is to develop an unsupervised feature learning method for activity recognition to eliminate the need for manual feature engineering, making it more accurate in learning the underlying features of the data. Furthermore, the proposed method maps the sensor data into a lower dimensionality feature space, consequently, reduces the computational cost and improves generalization. The proposed method is experimentally validated on a public dataset of 30 subjects. The results show that the proposed method able to effectively learn a compact and salient feature representation of the data and achieves high classification accuracy.

The remainder of this paper is organized as follows. Section 2 reviews the related works. In Sect. 3, we present the proposed methodology that consists of data collection, signal segmentation, feature learning method and activity recognition. Section 4 presents the experimental results and their discussion. Finally, the conclusions are presented in Sect. 5.

2 Related works

Numerous feature extraction methods have been proposed for activity recognition which can be divided into supervised feature extraction and unsupervised feature extraction. In supervised feature extraction, features are extracted from the segments of sensor data with known labels (labeled dataset). These features provide an intuitive representation of how a signal pattern is produced by an activity. The features are shown to be effective in classifying physical activities [6]. Features are also extracted from Fourier transformed signals such as spectral energy and entropy [7]. In [8], the decision tree algorithm is used to model time-domain and frequency-domain features such as mean, variance skew, kurtosis, spectral centroid and peak frequency. An artificial neural network is built to model the mean and standard deviation of each axis of the accelerometer for activity recognition [9]. The support vector machine is used as a classifier for offline and online activity recognition [10]. The features are standard deviation and minimum value extracted from acceleration and orientation angle measurements. The classification accuracies of the aforementioned studies are in the range of 0.850 to 0.950.

An ensemble model using a voting scheme is proposed for activity recognition [11]. The ensemble model consists of decision tree, logistic regression and neural network classifiers. Besides the commonly used features, features such as time between peaks, binned distribution are extracted from the signals. An ensemble classifier is proposed using the cascading method for activity recognition [12]. The model consists of extremely gradient boosting trees (XGBoost), random forest, extremely randomized trees and softmax regression. A two-stage hierarchical classifier is proposed using continuous hidden Markov models for activity recognition [13]. The first stage is to distinguish the activity data into dynamic activity and static activity, while the second stage further distinguishes it into the final activity class. In both studies, a more complex set of features is extracted from the signals such as autoregression coefficients, signal entropy and the angle between two vectors. Wavelet-based features were proposed for classifying dynamic activities. However, they were not as effective as time-domain and frequency-domain features [14].

Feature reduction method such as principal component analysis has been used to determine the projection direction of the most variation in the data. The projection maximizes the discriminability of the data to improve classification accuracy [15]. Random forest was used to model the reduced features. In [16], cepstral analysis is proposed to extract powerful discriminative features in the form of cepstral coefficients. The features were useful for distinguishing dynamic activities such as running, cycling and jumping. Single and ensemble classifiers such as support vector machine, neural network and random forest are used to model the features. Wang et al. utilized Ensemble Empirical Mode Decomposition (EEMD) to decompose time series data into several elemental signals called intrinsic mode function (IMF). Then, features such as mean crossing rate and autoregressive coefficients are extracted for activity recognition [17]. Although these hand-crafted features together with machine learning methods are effective in classifying different activities, they require expert knowledge and the task is difficult and laborious in practice.

Deep learning methods have been widely used to learn the features of the data automatically in various domains such as medical imaging analysis [18], video game playing [19] and cybersecurity [20]. In the domain of activity recognition, deep learning has also been extensively applied for classifying the sensor data into activities. In [21], a deep convolutional neural network (CNN) is proposed for classifying locomotion activities and body postures. The model consists of three layers of convolution and max-pooling operations for feature learning, producing 192 feature maps. Similar work is found in [22], whereby a CNN is proposed for online smartphone activity recognition. The feature learning pipeline consists of two convolutional layers followed by a max-pooling layer to produce 200 feature maps. In [23], a shallow CNN is proposed for real-time activity recognition. The proposed model consists of a single layer of convolution and max-pooling operations producing 196 feature maps. A two-stage deep learning model is proposed to improve the feature learning for activity recognition [24]. The first deep learning model is used to learn features of activity classes that are difficult to classify, while the second model is used to classify the other activity classes. A deep learning model using CNN and long short-term memory is proposed for activity recognition [25]. The proposed model consists of four convolutional layers for feature learning, producing 64 feature maps that are fed to recurrent dense layers to model the temporal dependencies of the features. Although CNNs have been shown to be effective in learning discriminative features for activity recognition, they are purely supervised learning methods which cannot leverage the unlabeled data.

The unsupervised feature extraction methods extract features from raw data without labels (or known as an unlabeled dataset). It is especially useful in the activity recognition application where labeled datasets are rare and difficult to obtain due to time-consuming annotation processes. Autoencoder is a type of unsupervised neural networks that can be used to learn feature representation of data. It learns the feature representation by training the network to reconstruct the data at the output layer. An improved autoencoder is proposed to effectively extract the features from the inertial sensor data for activity recognition [26]. The autoencoder allows the steepness of the activation function to be controlled, consequently reducing overfitting the data. In [27], a stacked denoising autoencoder is proposed to learn the feature representation of inertial sensor data for activity recognition. The stacked denoising autoencoder is capable of extracting more robust features by undoing the effect of the corruption process applied to the data. Then, the features are modeled using Light Gradient Boosting Machine for activity classification. A full neural network pipeline is proposed using stacked denoising autoencoder for feature learning and activity recognition [28]. However, one of the major drawbacks of the proposed models is the relatively long training time because the training is performed one layer at a time. In [29], an unsupervised feature learning method based on denoising autoencoder for activity recognition is proposed. A penalty term based on Kullback–Leibler divergence is added to the cost function to encode a salient feature representation. Although the aforementioned works extract effective features, the denoising autoencoders do not incorporate convolutional layers in the models. As a result, the local temporal structure is not sufficiently leveraged to learn the salient feature representation of the sensor data. The studies also do not consider transitional activities which are important to provide early preventive measures of fall accidents.

In this paper, by extending our preliminary work in [29], an unsupervised feature learning pipeline based on convolutional denoising autoencoder (CDAE) is proposed. The CDAE incorporates convolutional and pooling layers to maximally leverage the local temporal structure of the data for feature extraction. In addition, the CDAE is jointly trained with a CNN to force the encoding layer to encode a more salient feature representation of the data. As a result, more discriminative features can be extracted to achieve high classification accuracy. Furthermore, the proposed method produces a compact feature representation by removing unimportant characteristics of the data. This reduces overfitting and improves the generalization of the classification model. Unlike previous works, this work also deals with various lengths of transitional activity signals in activity recognition. The proposed method is evaluated using a public dataset. In addition, we have implemented deep learning models that can automatically extract features to compare with the proposed method. The experimental results show that the proposed method can achieve a high classification accuracy and its performance is comparable to the deep learning models.

3 Proposed methodology

The proposed methodology consists of the following phases: data collection, signal segmentation, unsupervised feature learning and activity classification as shown in Fig. 1. In the data collection, a wearable device embedded

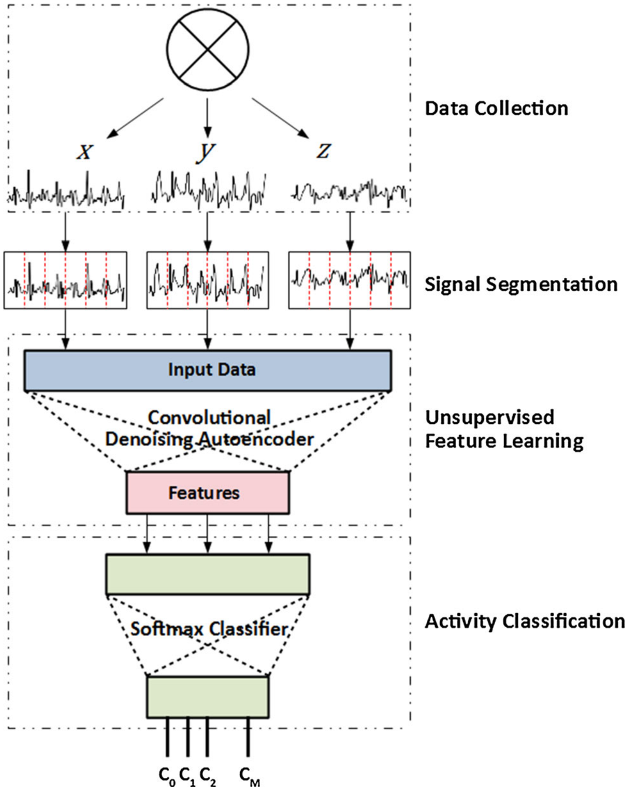


Fig. 1 The block diagram of the proposed method

with an inertial sensor records the measurement of body motion of the subjects while performing the activities of interest. The wearable device is attached to the waist of the subjects to acquire the major body movement [30]. The activities are the four basic activities which are walking, standing, sitting and lying down, and also the transitions between the body postures such as stand-to-sit, sit-to-stand, sit-to-lie and lie-to-sit. The following sections explain the key phases of the proposed methodology.

3.1 Signal segmentation

In activity recognition, the signals need to be segmented into windows for classification. The most widely used signal segmentation method is the fixed sliding window whereby the signals are segmented into windows of a fixed-size with a degree of overlap [31]. It should be noted that the size of the window segmentation defines the size of the input data of the CDAE. However, one challenge of fixed sliding window is to determine the optimum window size. A short window size could truncate the signals into multiple windows, while a larger window could contain multiple activity signals. In both cases, misclassification could happen especially for transitional activities where the length of transitional activity signals varies depending on the time to complete the activity [32]. In order to overcome the limitation of the fixed sliding window method, we use the adaptive sliding window for the signal segmentation [33]. The adaptive sliding window defines an initial window size that can be expanded to accommodate activity signals that are longer than the initial window. The size of the expansion is defined by the expansion factor. Therefore, the transitional activity signals can be segmented according to the length of the signals. Then, the linear interpolation is used to resize the windows with a different number of samples to the size of the initial window. Let s_i denote a window segmentation i which represents an input vector of the proposed model. A series of window segmentation, S, is given as follows.

$$S = \left[\boldsymbol{x}_0^c, \boldsymbol{x}_1^c, \ldots, \boldsymbol{x}_N^c\right] \tag{1}$$

where N is the number of window segmentation. A motion sensor such as a triaxial accelerometer generates three values which represent the measurement along different axes, X-axis, Y-axis and Z-axis. Let the input channels are denoted by c. The three axes data from a triaxial accelerometer are given as follows.

$$\begin{aligned} \boldsymbol{x}_i^x &= \left[x_j^x, x_{j+1}^x, \ldots, x_{j+K-1}^x\right] \\ \boldsymbol{x}_i^y &= \left[x_j^y, x_{j+1}^y, \ldots, x_{j+K-1}^y\right] \\ \boldsymbol{x}_i^z &= \left[x_j^z, x_{j+1}^z, \ldots, x_{j+K-1}^z\right] \end{aligned} \tag{2}$$

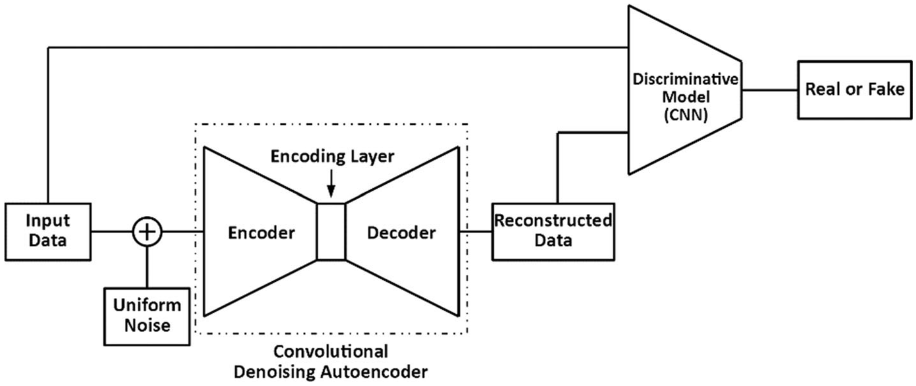
where K is the size of the initial window segmentation. The input shape of the CDAE is $K \times c$.

3.2 Unsupervised feature learning

The unsupervised feature learning is based on denoising autoencoder which is a variant of autoencoder. The block diagram of the proposed method is given in Fig. 2. Autoencoders learn a feature representation of data under reconstruction tasks using unlabeled data. Autoencoder consists of encoder and decoder whereby the encoder compresses the input data by propagating the data from one hidden layer to the subsequent layers. This process results in a feature representation encoded in the encoding layer. The decoder takes the encoded features to reconstruct the input data. Since a constraint is imposed on the encoding layer such as limiting the number of neurons, the autoencoders are forced to learn the salient feature representation of the data. In order to improve the reconstruction task, a discriminative model is integrated into the unsupervised feature learning pipeline whereby the discriminative model is fed with the reconstructed data and input data and learned to distinguish between the real and fake signals.

A denoising autoencoder attempts to learn a robust feature representation by introducing stochastic noise to the input data, and the autoencoder is required to reconstruct

Fig. 2 The unsupervised feature learning pipeline



the data from corrupted data [34]. The process of denoising the data allows a more effective feature representation to be learned by the autoencoder. In this research, the corrupted input signal, $\acute{\boldsymbol{x}}_i^c$ is obtained by adding uniform random noise, $\boldsymbol{\omega}$ to the input signal.

$$\acute{\boldsymbol{x}}_i^c = \boldsymbol{x}_i^c + \boldsymbol{\omega} \tag{3}$$

Traditionally, denoising autoencoders are built using fully connected layers. In this research, we built the denoising autoencoder using 1D convolutional and 1D max-pooling layers due to their ability to handle time series data [35]. The 1D convolution operation leverages the local temporal structure of the signals, and the 1D max-pooling operation makes the network invariance to small translations in the input signal [36]. The 1D convolutional and 1D max-pooling layers are stacked alternately, to extract features in a hierarchical manner whereby the initial layers extract the primitive features while the deeper layers extract complex ones by combining the extracted basic features. As a result, a salient feature representation can be learned by the CDAE.

The CDAE was designed with the aim to provide a small set of features for activity classification. We designed two architectures of CDAEs, and the parameters of the CDAEs are given in Tables 1 and 2. The encoders consist of 1D convolutional and 1D max-pooling layers which are stacked after each other. The kernel size of the first 1D convolutional layer is 11, and the kernel size is decreasing as the network gets deeper to the last 1D convolutional layer which employs a kernel of 3 (CDAE A) and 5 (CDAE B). The input of each 1D convolutional layer is padded with zeros. Thus, the output shape is the same as the input shape. To reduce the size of the outputs by half, the kernel size of the 1D max-pooling layers is set to 2. The first 1D convolutional layer has a depth value (number of feature maps) of 10, and the depth value is progressively increased to 50 (CDAE A) and 40 (CDAE B) as the network gets deeper to the last 1D convolutional layer. The number of encoded features is defined by the output shape at the encoding layer which is determined by the parameters of the 1D convolution and 1D max-pooling operations. The encoding layer of CDAE A and CDAE B encodes features of size $3 \times d$ and $6 \times d$, where d is the depth of the encoding layer or specifically the number of kernels at this layer. All 1D convolutional layers use exponential linear unit (ELU) activation function. The layer structure of the decoder is symmetric to the encoder. Up-sampling layers are used to increase the size of the outputs. The deep model

Table 1 The architecture and parameters of CDAE A

	Layer	Kernel size	Stride	Output shape
1	Input			100 × 3
2	Conv	11	1	100 × 10
3	Max-pool	2	2	50 × 10
4	Conv	9	1	50 × 20
5	Max-pool	2	2	25 × 20
6	Conv	7	1	25 × 30
7	Max-pool	2	2	12 × 30
8	Conv	5	1	12 × 40
9	Max-pool	2	2	6 × 40
10	Conv	3	1	6 × 50
11	Max-pool	2	2	3 × 50
12	Conv	3	1	3 × d
13	Conv	3	1	3 × 50
14	Up-sample	2	2	6 × 50
15	Conv	3	1	6 × 40
16	Up-sample	2	2	12 × 40
17	Conv	5	1	12 × 30
18	Up-sample	2	2	25 × 30
19	Conv	7	1	25 × 40
20	Up-sample	2	2	50 × 40
21	Conv	9	1	50 × 50
22	Up-sample	2	2	100 × 50
23	Conv	11	1	100 × 3

Layer 1–11 represent the encoder. Layer 12 is the encoding layer. The decoder is symmetric to the encoder in terms of the layer structure

Table 2 The architecture and parameters of CDAE B

	Layer	Kernel size	Stride	Output shape
1	Input			100 × 3
2	Conv	11	1	100 × 10
3	Max-pool	2	2	50 × 10
4	Conv	9	1	50 × 20
5	Max-pool	2	2	25 × 20
6	Conv	7	1	25 × 30
7	Max-pool	2	2	12 × 30
8	Conv	5	1	12 × 40
9	Max-pool	2	2	6 × 40
10	Conv	3	1	6 × d
11	Conv	3	1	6 × 40
12	Up-sample	2	2	12 × 40
13	Conv	5	1	12 × 30
14	Up-sample	2	2	25 × 30
15	Conv	7	1	25 × 20
16	Up-sample	2	2	50 × 20
17	Conv	9	1	50 × 10
18	Up-sample	2	2	100 × 10
19	Conv	11	1	100 × 3

Layer 1–9 represent the encoder. Layer 10 is the encoding layer. The decoder is symmetric to the encoder in terms of the layer structure

allows the features to be learned in a hierarchical manner, resulting in more discriminative features to be extracted from the data.

The CDAEs are trained by regressing to the original input signals. Specifically, we train the CDAEs to minimize the reconstruction loss or the mean squared error between the input signals and the reconstructed signals. However, the reconstruction task is inherently multimodal, and our experiments demonstrate that the autoencoder could not reconstruct the shape of the signals correctly as shown in Fig. 3 (top). As can be seen in the figure, the magnitude of the reconstructed signals varies and far from the original signals. This indicates that the CDAE fails to learn the underlying structure of the data, resulting in a non-discriminative feature representation to be encoded. To address this issue, we propose to integrate a discriminative model into the unsupervised feature learning pipeline (as shown in Fig. 2) and jointly train the networks to minimize the reconstruction loss and adversarial loss [37]. The discriminative model takes the reconstructed signals and the input signals and tries to distinguish the real from the fake signals. This allows the CDAEs to learn to reconstruct the signals as real as possible. Let $\boldsymbol{x}$ denote the multi-channel input signals, and $\hat{\boldsymbol{x}}$ is the corresponding reconstructed signals. The cost function is given as

$$J = L_r(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda L_{\mathrm{a}} \tag{4}$$

where L_{r} is the reconstruction loss which is defined as

$$L_{\mathrm{r}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sum_{j=0}^{K} \left(x_j - \hat{\boldsymbol{x}}_j\right)^2 \tag{5}$$

λ is the scaling factor of adversarial loss, and L_{a} is the adversarial loss which is defined as

$$L_{\mathrm{a}} = -[\ln(D(\boldsymbol{x})) + \ln(1 - D(\hat{\boldsymbol{x}}))] \tag{6}$$

where D is the discriminative model. We build a convolutional neural network with two layers of 1D convolution and max-1D pooling operations followed by two hidden fully connected layers. The output layer has two nodes corresponding to real and fake signals. All 1D convolutional layers use ELU activation function. Figure 3 (bottom) shows a reconstructed signal by the CDAE trained with the discriminative model.

3.3 Activity classification

Once the CDAE is trained, the discriminative model is removed, and the decoder is replaced with a softmax classifier. The encoder provides a set of features to the classifier for activity recognition as shown in Fig. 4. Here, we consider supervised fine-tuning where the network is trained to minimize the prediction error of activity classification. During training, the layers of the encoder are frozen, and the weights of the classifier are fine-tuned. The activity classes are used as targets. The classifier is built using two hidden fully connected layers with ELU activation function followed by a softmax output layer. The source code of the proposed model can be found in Github[1].

4 Experiments and results

4.1 Experimental setup

We performed the experiments using a public dataset [38]. The dataset contains activity signals collected from 30 subjects using a smartphone inertial sensor. The smartphone was attached to the front waist. The dataset includes basic activities such as walking, standing, sitting, lying down and the transitions between two postures. Table 3 shows the number of samples for each activity class. The inertial sensor data were normalized to the range of − 1 and 1. The initial window size, K, was set to 100, and both the overlapping and expansion factors were set to 0.5. The dataset was split into training part with data from 20 subjects and test part with data from the remaining 10 subjects.

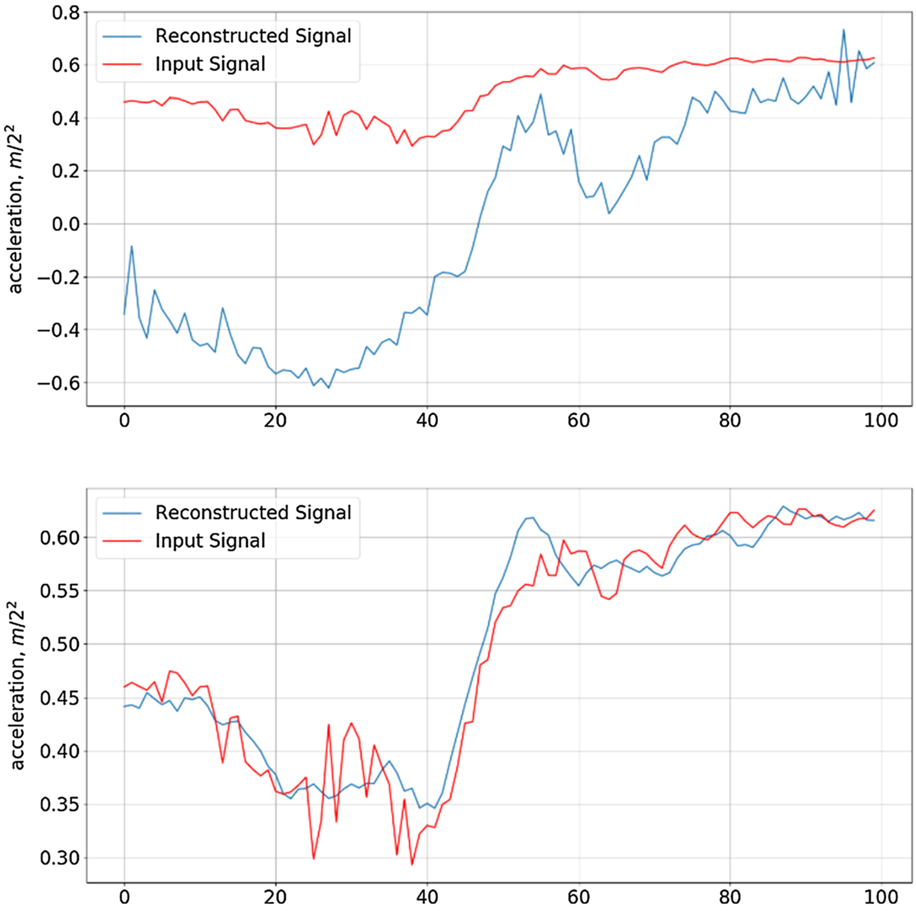


Fig. 3 The reconstructed signals without (top) and with (bottom) jointly trained CDAE

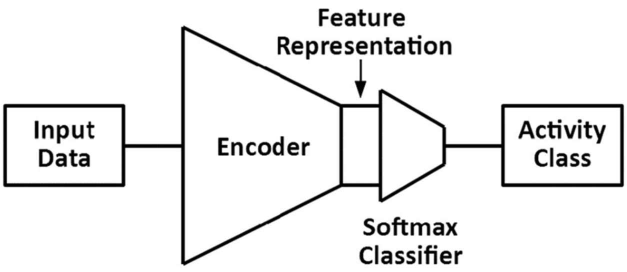


Fig. 4 The activity classification pipeline

Table 3 Number of samples in different classes

Activity	Number of samples
Walking	1176
Standing	1533
Sitting	1329
Lying down	1385
Stand-to-sit	60
Sit-to-stand	60
Sit-to-lie	60
Lie-to-sit	60

Corrupt training data were created by adding uniform random noise in the interval of $(-0.05, 0.05)$ to the training data. The CDAE was trained on 80 batches of 16 input signals for 1000 epochs using Adam optimizer. Early stopping and dropout techniques were used to prevent overfitting the training data. The learning rate and λ were set to 0.001. We performed two experiments of activity recognition. The first experiment is activity recognition with accelerometer data only and the second experiment using both accelerometer and gyroscope data.

The classifier of the network was fine-tuned using the same approach. But this time, the original training data were fed to the network and the activity classes were set as the target. Adam optimizer was used to fine-tune the weights, and early stopping and dropout techniques were used to prevent overfitting the data. We evaluated the performance of the proposed method using recall, precision, F-score and accuracy. Recall is defined as the ability of the classifier to identify the activity class of a window segmentation. Precision reflects the ability of the classifier to distinguish an activity class from all the other classes. F-score is the average of recall and precision. Accuracy is the fraction of correctly classified window segmentation. The evaluation metrics are given by

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$F\text{-score} = \frac{2 \times \text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})} \tag{9}$$

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

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

4.2 Autoencoder architecture

We analyze the performance of both CDAE architectures. We experimented with depth, $d = \{1, 2, 3, 4, 5, 6, 7\}$ in order to determine the best architecture for feature learning. Figures 5 and 6 illustrate the classification accuracy variation versus the increase in d for both experiments. The value of d is the number of kernels at the encoding layer. Hence, d determines the number of features for activity recognition. Therefore, it has a significant influence on classification accuracy. As can be seen in Figs. 5 and 6, the classification accuracy is relatively lower when the value of d is less or equals 3 and stabilizes beyond 3. Figure 5 also shows that increasing the value of d does not necessarily lead to an increase in classification accuracy. In the first experiment, the highest classification accuracy is 0.908 for CDAE A and 0.907 for CDAE B, while in the second experiment, the highest classification accuracy is 0.897 for CDAE A and 0.914 for CDAE B. Thus, $d = 5$ and $d = 4$ are proposed for CDAE A and CDAE B, respectively, for activity recognition with accelerometer only. For activity recognition with accelerometer and gyroscope, $d = 6$ and $d = 7$ are proposed for CDAE A and CDAE B.

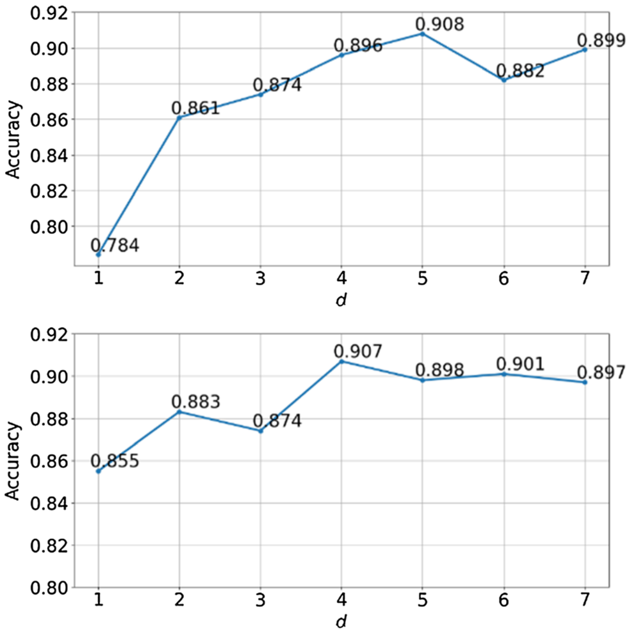


Fig. 5 Accuracy of activity recognition of CDAE A (top) and CDAE B (bottom) using accelerometer only

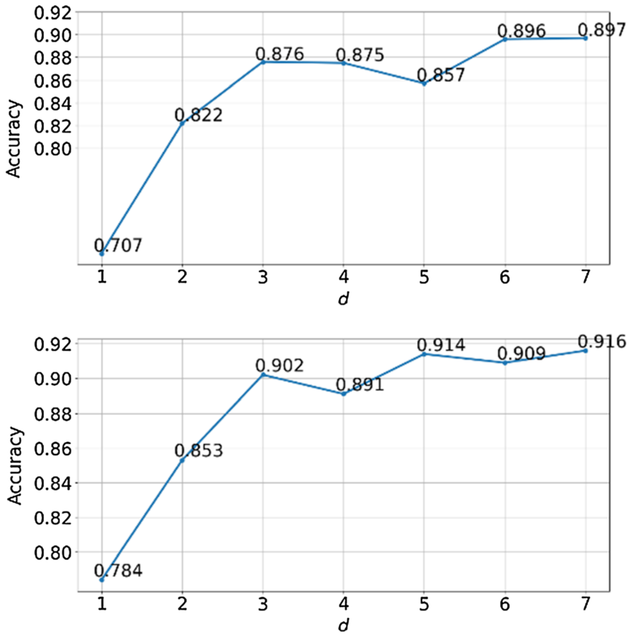


Fig. 6 Accuracy of activity recognition of CDAE A (top) and CDAE B (bottom) using both accelerometer and gyroscope

4.3 Classification performance

The classification performance of the proposed CDAEs in classifying each activity is analyzed. We have additionally performed a threefold user-based cross-validation to validate the classification performance. In general, the proposed CDAEs performed well in classifying most of the activities considered in the experiments. Stand-to-sit (A3) and sit-to-stand (A5) activities have relatively lower performance than other activities for both CDAEs, and it is observed that most of the misclassified samples are classified as walking activity (A1). This is because the activities share similar characteristics where they involve dynamic movement in an upright position. Standing activity (A2) is also easily misclassified as sitting activity (A4) and vice versa. As can be seen in Tables 4 and 5, 8.8% and 7.9% of standing samples are misclassified as sitting, and 9.9% and 7.1% of sitting samples are misclassified as standing. Tables 6 and 7 show that 9.0% and 7.9% of standing samples are misclassified as sitting, and 8.1% and 6.3% of sitting samples are misclassified as standing. The reason is that both activities have similar signal patterns, and as a result, the features that have been learned by the autoencoders have similar representation.

Table 4 Confusion matrix of activity recognition using accelerometer only (Autoencoder A)

	A1	A2	A3	A4	A5	A6	A7	A8
A1	1085	34	10	1	50	0	0	0
A2	36	1276	1	127	2	0	0	0
A3	8	2	47	1	1	1	0	0
A4	2	125	0	1124	1	0	0	0
A5	8	1	1	0	50	0	0	0
A6	0	0	0	1	0	55	4	0
A7	0	0	0	1	0	1	1294	5
A8	1	0	0	0	5	0	1	53

The activities are labeled as A1 (walking), A2 (standing), A3 (stand-to-sit), A4 (sitting), A5 (sit-to-stand), A6 (sit-to-lie), A7 (lying down) and A8 (lie-to-sit)

Table 5 Confusion matrix of activity recognition using accelerometer only (Autoencoder B)

	A1	A2	A3	A4	A5	A6	A7	A8
A1	1114	34	10	5	12	5	0	0
A2	39	1282	0	115	5	1	0	0
A3	5	0	53	0	0	2	0	0
A4	0	90	2	1157	2	0	1	0
A5	5	0	1	0	54	0	0	0
A6	0	0	3	0	0	54	3	0
A7	0	0	0	25	0	2	1274	0
A8	1	0	0	0	0	0	0	59

The activities are labeled as A1 (walking), A2 (standing), A3 (stand-to-sit), A4 (sitting), A5 (sit-to-stand), A6 (sit-to-lie), A7 (lying down) and A8 (lie-to-sit)

Table 6 Confusion matrix of activity recognition using accelerometer and gyroscope (Autoencoder A)

	A1	A2	A3	A4	A5	A6	A7	A8
A1	1114	16	21	6	15	6	1	1
A2	61	1242	2	130	4	1	2	0
A3	14	1	41	0	1	1	0	2
A4	2	102	4	1138	4	1	0	1
A5	11	1	2	0	41	1	0	4
A6	2	0	1	0	0	56	0	1
A7	1	0	2	0	0	1	1296	1
A8	3	0	0	1	7	0	2	47

The activities are labeled as A1 (walking), A2 (standing), A3 (stand-to-sit), A4 (sitting), A5 (sit-to-stand), A6 (sit-to-lie), A7 (lying down) and A8 (lie-to-sit)

Table 7 Confusion matrix of activity recognition using accelerometer and gyroscope (Autoencoder B)

	A1	A2	A3	A4	A5	A6	A7	A8
A1	1132	26	4	1	15	0	0	2
A2	56	1265	1	115	5	0	0	0
A3	15	0	40	0	2	3	0	0
A4	1	79	1	1170	0	1	0	0
A5	13	0	1	0	42	0	0	4
A6	0	0	0	0	0	58	0	2
A7	0	0	0	0	0	5	1294	2
A8	2	0	0	0	2	0	0	56

The activities are labeled as A1 (walking), A2 (standing), A3 (stand-to-sit), A4 (sitting), A5 (sit-to-stand), A6 (sit-to-lie), A7 (lying down) and A8 (lie-to-sit)

The recall, precision and F-score measures of each activity for both CDAEs are given in Figs. 7 and 8. Overall, the highest classification accuracies are achieved by CDAE B which are 0.932 and 0.934. Table 8 shows a comparison of the classification accuracy of the proposed method.

We have also analyzed the influence of kernel size on the performance of activity recognition. A smaller kernel size allows the CDAE to learn highly local and complex features in the data, while a larger kernel size means a larger receptive field thereby more generic features will be learned. However, the CDAE will have more information to capture the trend of the data. This experiment was carried out by varying the kernel sizes of the convolutional

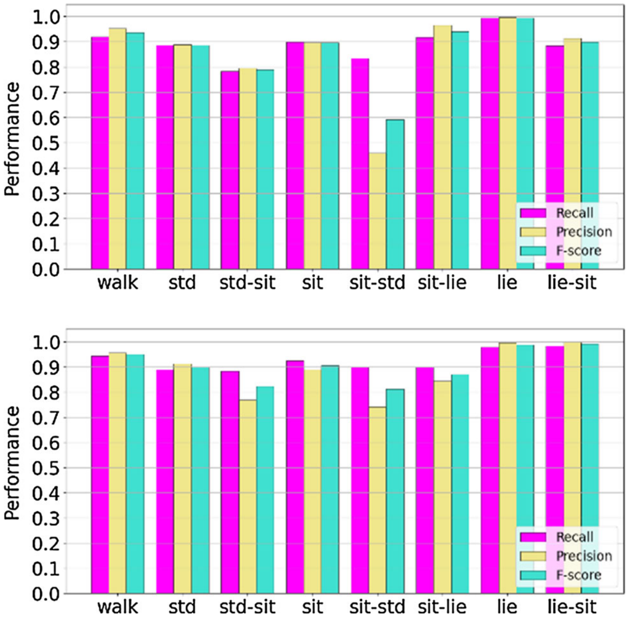


Fig. 7 The recall, precision and F-score measures of CDAE A (top) and CDAE B (bottom) using accelerometer only

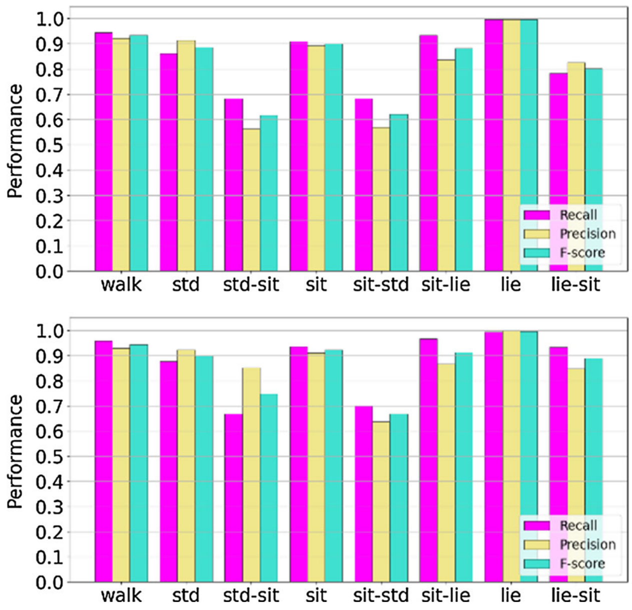


Fig. 8 The recall, precision and F-score measures of CDAE A (top) and CDAE B (bottom) using accelerometer and gyroscope

Table 8 Comparison of classification accuracies of the proposed method

Architecture	Classification accuracy
CDAE A (Accelerometer only)	0.920
CDAE B (Accelerometer only)	0.932
CDAE A (Accelerometer and gyroscope)	0.919
CDAE B (Accelerometer and gyroscope)	0.934

layers of CDAE B. Tables 9 and 10 show the classification accuracy of CDAE B with different kernel sizes. It is found that the kernel size does not have a significant influence on the performance of activity recognition. Using smaller kernel sizes, the classification accuracy is decreased slightly. However, a smaller kernel size reduces the number of weights and consequently reduction in computational cost.

Table 9 Comparison of classification accuracy of CDAE B using accelerometer only with different kernel sizes

Kernel size	Classification accuracy
11, 9, 7, 5, 3	0.932
9, 7, 5, 3, 3	0.927
7, 5, 3, 3, 3	0.920

Table 10 Comparison of classification accuracy of CDAE B using accelerometer and gyroscope with different kernel sizes

Kernel size	Classification accuracy
11, 9, 7, 5, 3	0.934
9, 7, 5, 3, 3	0.926
7, 5, 3, 3, 3	0.916

4.4 Comparison with machine learning models

We compare the performance of the proposed method with the conventional method whereby the hand-crafted features are extracted and modeled using machine learning techniques. In this work, we extract the commonly used features such as mean, variance, mean crossing rate and signal magnitude area [30]. We also extract absolute mean trend and mean trend features to capture the local trend of the signal within the segmentation which has shown to be effective in classifying the transitional activities [33]. The features are extracted from the three axes of acceleration and gyroscope data and also from the derived acceleration data that represent the sagittal plane, transverse plane and frontal plane. A total of 54 features have been extracted from the data. Then, feature selection is carried out using filter method based on the ANOVA *F*-value to rank the features with respect to their relevance. Three single classifiers (*K*-nearest neighbors, decision tree and support vector machine) and four ensemble classifiers (AdaBoost, Stacking, Random Forest and XGBoost) are built for the activity recognition. In order to determine the best set of features, first, the classifiers are modeled using the top 10% features. Then, the top 20% features are used to model the classifiers, and the experiments are repeated until the top 90% features. The set of features that produces the best classification accuracy is selected. From the experimental results, it is observed that the classification accuracies achieved by the machine learning models are lower than the proposed method. As shown in Table 11, the highest classification accuracies are achieved by XGBoost and Random Forest which are 0.904 (accelerometer only) and 0.922 (accelerometer and gyroscope), respectively.

4.5 Comparison with deep learning models

We compare the performance of the proposed method with deep learning convolutional neural networks. The deep learning models that were built have four convolution and max-pooling layers followed by flattening operation, two

Table 11 Comparison of classification accuracy of machine learning models

Accelerometer		Accelerometer and gyroscope	
Machine learning model	Classification accuracy	Machine learning model	Classification accuracy
K-nearest neighbors	0.863	*K*-nearest neighbors	0.912
Decision tree	0.854	Decision tree	0.843
Support vector machine	0.867	Support vector machine	0.898
AdaBoost	0.871	AdaBoost	0.908
Stacking	0.863	Stacking	0.914
Random forest	0.858	Random forest	0.922
XGBoost	0.904	XGBoost	0.921

fully connected layers and the softmax output layer. The deeper layers of the models were designed to have more kernels (depth) than the initial layers. This will allow the networks to learn a large number of high-level features for activity recognition. We built two deep learning models in which both models have the same number of layers. Each layer has the same kernel size and number of neurons but with different depth values at the convolutional layers. The architecture of the deep learning models is given in Table 12. Initially, we set the depth values to 5, 10, 15 and 20 (the model is referred to as M1). The model is trained with cross-entropy loss using Adam optimizer. The classification accuracy using accelerometer only is 0.931 which is 0.001 lower than the proposed method while classification accuracy using both accelerometer and gyroscope is 0.932 which is 0.002 lower than the proposed method. Then, the depth values of the convolutional layers are increased to 10, 20, 30 and 40 (the model is referred to as M2) in order to learn more features. The model is trained using the same approach. It is observed that the classification accuracy of the model is increased to 0.941 for both experiments which is slightly higher than the proposed method.

Although the deep learning model achieved a slightly higher accuracy, it requires a large feature vector for feature learning and classification. As shown in the experiment, model M2 requires a feature vector of 240 (6×40). This results in the increase in feature dimension which in turn increases the computational cost of the model. In comparison, the depth of the proposed CDAE is relatively much smaller with the depth value of 4 and 7 only at the encoding layer, resulting in a feature vector of 24 (6×4) and 42 (6×7), respectively. This allows the feature dimension and computational cost of the model to be reduced significantly. Furthermore, unlike convolutional neural networks where fully connected layers are used as a classifier, the proposed method allows other preferred machine learning algorithms such as decision tree, Naïve

Table 12 The architecture of the deep learning models

	Layer	Kernel size	Depth (d) M1/M2	Stride	Output shape
1	Input		3		100 × 3
2	Conv1D	9	5/10	1	100 × d
3	Max-pool	2		2	50 × d
4	Conv1D	7	10/20	1	50 × d
5	Max-pool	2		2	25 × d
6	Conv1D	5	15/30	1	25 × d
7	Max-pool	2		2	12 × d
8	Conv1D	3	20/40	1	12 × d
9	Max-pool	2		2	6 × d
10	Flatten				

	Layer	Unit M1/M2	Output shape
11	FC	60/120	60/120
12	FC	30/60	30/60
13	FC (output)	10	10

FC fully connected

Bayes or support vector machine to be used to model the learned features for classification. These lightweight classifiers could reduce the number of parameters and consequently the model size. The performance of the deep learning models in terms of recall, precision and F-score is given in Figs. 9 and 10. Table 13 shows the comparison of classification accuracy of the deep learning models.

4.6 Comparison with state-of-the-art methods

In comparison with the state-of-the-art methods, the proposed method appears promising in several regards. Table 14 reports a summary of the state-of-the-art methods. First, its performance is comparable if not better with the previously proposed methods. In [26], a deep belief network consists of two autoencoders in sequence followed by one backpropagation layer is proposed for activity recognition. The autoencoders accept the features extracted from the signals as input, compress and reconstruct the input data to produce a set of features that will be used to classify the activities. The proposed method is evaluated on a dataset consisting of measurements of inertial and magnetic sensors attached to five different parts of the body. Classification accuracy of 0.949 is achieved which is slightly higher than our proposed method's accuracy. However, the proposed method utilizes an additional sensor and multiple sensor attachment which might impede the activity of daily living.

In [28], a full neural network pipeline is proposed using a stacked denoising autoencoder for activity recognition. The stacked denoising autoencoder consists of two hidden fully connected layers with 1000 units in each layer. The square error loss with Kullback–Leibler divergence as the regularizer is used to train the network. The experiments were carried out on a dataset collected from 12 subjects, each carrying a mobile phone integrated with accelerometer, gyroscope, magnetometer and barometer sensors. The proposed model recorded the highest classification accuracy of 0.940. It is noted that the classification accuracy is reduced to 0.925 when activity recognition is performed using two sensors (accelerometer and barometer). To further improve the performance, the network parameters are fine-tuned by increasing the number of hidden units to 1500. The classification accuracy is slightly improved by 0.003.

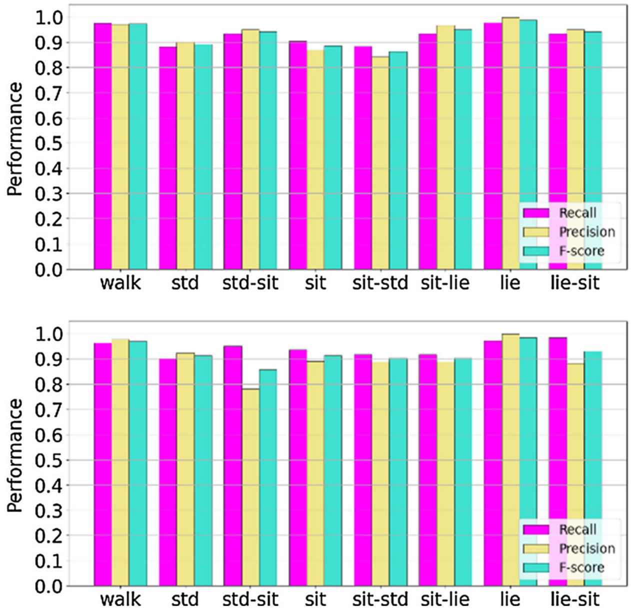


Fig. 9 The recall, precision and F-score measures of model M1 (top) and model M2 (bottom) using accelerometer

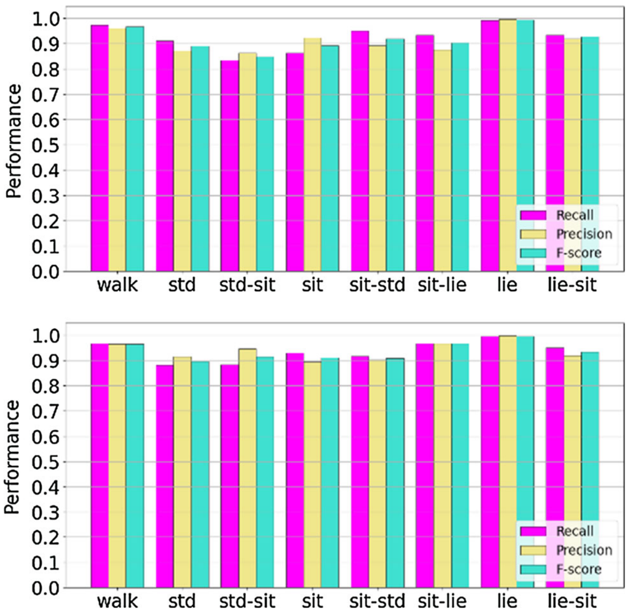


Fig. 10 The recall, precision and F-score measures of model M1 (top) and model M2 (bottom) using accelerometer and gyroscope

Table 13 Comparison of classification accuracy of the deep learning models

Architecture	Classification accuracy
Model M1 (Accelerometer only)	0.931
Model M1 (Accelerometer and gyroscope)	0.932
Model M2 (Accelerometer only)	0.941
Model M2 (Accelerometer and gyroscope)	0.941

In [27], a stacked denoising autoencoder is proposed to extract relevant features from activity signals. Then, the

Table 14 Classification accuracy and description of state-of-the-art methods

Relevant study	Classification accuracy	Description
Wang [26]	0.949	Activity recognition system with multiple wearable sensors
Gao et al. [27]	0.943	The classification accuracy dropped when two sensor measurements were used in activity recognition
Gu et al. [28]	0.982	The experiments were conducted by separating the transitional activity classes from the basic activity classes
Proposed method	0.934	Use of convolutional layers with max-pooling layers to learn a more salient feature representation

light gradient boosting is used to model the features for activity recognition. The stacked denoising autoencoder consists of three hidden fully connected layers with 400 units in the first hidden layer, 200 units in the second hidden layer and 40 units in the third hidden layer. The mean square error loss is used to train the network. The proposed model is evaluated on four datasets. The first dataset contains inertial magnetic and air pressure data, and the classification accuracy is recorded at 0.957. The second dataset is derived from the first dataset by removing the air pressure sensor. The recorded classification accuracy is reduced to 0.937. The third dataset contains only dynamic and static activities, while the fourth dataset contains only transitional activities. Both datasets were collected using an inertial sensor only. The proposed models recorded the highest classification accuracies of 0.982 and 0.963. It has to be noted that unlike our proposed method, the experiments were conducted by separating the transitional activity classes from the basic activity classes. Therefore, it is difficult to conclude if the models actually perform better. Furthermore, the local temporal structure was not leveraged because the convolutional layer was not incorporated in the proposed autoencoders [26–28].

5 Conclusion

In this paper, we propose an unsupervised feature learning method based on convolutional denoising autoencoder for automatically extracting discriminative features from triaxial accelerometer data to eliminate the need for manual feature engineering. In the proposed method, the convolutional and pooling layers are exploited to maximally leverage the local structure of the data. Furthermore, we propose a joint training approach to enhance the reconstruction task. As a result, the proposed method able to learn a more salient feature representation of the data. Experimental results show that the proposed method achieves a higher classification accuracy than the machine learning-based techniques. In comparison with deep learning models and state-of-the-art methods, the proposed method achieves a comparable classification accuracy without requiring a large number of features.

Acknowledgements This work has been supported in part by the Universiti Sains Malaysia under Short-Term Grant 304/PKOMP/6315206.

Compliance with ethical standards

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

References

1. Landi F, Onder G, Carpenter I et al (2007) Physical activity prevented functional decline among frail community-living elderly subjects in an international observational study. J Clin Epidemiol 60:518–524. https://doi.org/10.1016/j.jclinepi.2006.09.010
2. Jansen FM, Prins RG, Etman A et al (2015) Physical activity in non-frail and frail older adults. PLoS ONE 10:e0123168. https://doi.org/10.1371/journal.pone.0123168
3. Durstine JL, Gordon B, Wang Z, Luo X (2013) Chronic disease and the link to physical activity. J Sport Health Sci 2:3–11. https://doi.org/10.1016/j.jshs.2012.07.009
4. Naci H, Ioannidis JPA (2013) Comparative effectiveness of exercise and drug interventions on mortality outcomes: metaepidemiological study. BMJ 347:f5577. https://doi.org/10.1136/bmj.f5577
5. Lara OD, Labrador MA (2013) A survey on human activity recognition using wearable sensors. IEEE Commun Surv Tutor 15:1192–1209. https://doi.org/10.1109/SURV.2012.110112.00192
6. Suto J, Oniga S, Lung C, Orha I (2018) Comparison of offline and real-time human activity recognition results using machine learning techniques. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3437-x
7. Rosati S, Balestra G, Knaflitz M (2018) Comparison of different sets of features for human activity recognition by wearable sensors. Sensors 18:4189. https://doi.org/10.3390/s18124189
8. Parkka J, Ermes M, Korpipaa P et al (2006) Activity classification using realistic data from wearable sensors. IEEE Trans Inf Technol Biomed 10:119–128. https://doi.org/10.1109/TITB.2005.856863
9. Kwon M-C, Choi S (2018) Recognition of daily human activity using an artificial neural network and smartwatch. Wirel

Commun Mob Comput 2018:2618045. https://doi.org/10.1155/2018/2618045

10. Fuentes D, Gonzalez-Abril L, Angulo C, Ortega JA (2012) Online motion recognition using an accelerometer in a mobile device. Expert Syst Appl 39:2461–2465. https://doi.org/10.1016/j.eswa.2011.08.098
11. Catal C, Tufekci S, Pirmit E, Kocabag G (2015) On the use of ensemble of classifiers for accelerometer-based activity recognition. Appl Soft Comput 37:1018–1022. https://doi.org/10.1016/j.asoc.2015.01.025
12. Xu S, Tang Q, Jin L, Pan Z (2019) A cascade ensemble learning model for human activity recognition with smartphones. Sensors. https://doi.org/10.3390/s19102307
13. Ronao CA, Cho S-B (2017) Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models. Int J Distrib Sens Netw 13:1550147716683687. https://doi.org/10.1177/1550147716683687
14. Preece SJ, Goulermas JY, Kenney LPJ, Howard D (2009) A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. IEEE Trans Biomed Eng 56:871–879. https://doi.org/10.1109/TBME.2008.2006190
15. Balli S, Sağbaş EA, Peker M (2018) Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm. Meas Control. https://doi.org/10.1177/0020294018813692
16. Vanrell SR, Milone DH, Rufiner HL (2018) Assessment of homomorphic analysis for human activity recognition from acceleration signals. IEEE J Biomed Health Inform 22:1001–1010. https://doi.org/10.1109/JBHI.2017.2722870
17. Wang Z, Wu D, Chen J et al (2016) A triaxial accelerometer-based human activity recognition via EEMD-based features and game-theory-based feature selection. IEEE Sens J 16:3198–3207. https://doi.org/10.1109/JSEN.2016.2519679
18. Ker J, Wang L, Rao J, Lim T (2018) Deep learning applications in medical image analysis. IEEE Access 6:9375–9389. https://doi.org/10.1109/ACCESS.2017.2788044
19. Justesen N, Bontrager P, Togelius J, Risi S (2019) Deep learning for video game playing. IEEE Trans Games. https://doi.org/10.1109/TG.2019.2896986
20. Xin Y, Kong L, Liu Z et al (2018) Machine learning and deep learning methods for cybersecurity. IEEE Access 6:35365–35381. https://doi.org/10.1109/ACCESS.2018.2836950
21. Ronao CA, Cho S-B (2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244. https://doi.org/10.1016/j.eswa.2016.04.032
22. Almaslukh B, Al Muhtadi J, Artoli AM (2018) A robust convolutional neural network for online smartphone-based human activity recognition. J Intell Fuzzy Syst 35:1609–1620. https://doi.org/10.3233/JIFS-169699
23. Ignatov A (2018) Real-time human activity recognition from accelerometer data using convolutional neural networks. Appl Soft Comput 62:915–922. https://doi.org/10.1016/j.asoc.2017.09.027
24. Huang J, Lin S, Wang N et al (2019) TSE-CNN: a two-stage end-to-end CNN for human activity recognition. IEEE J Biomed Health Inform. https://doi.org/10.1109/JBHI.2019.2909688
25. Ordóñez FJ, Roggen D (2016) Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 16:115. https://doi.org/10.3390/s16010115
26. Wang L (2016) Recognition of human activities using continuous autoencoders with wearable sensors. Sensors 16:189. https://doi.org/10.3390/s16020189
27. Gao X, Luo H, Wang Q et al (2019) A human activity recognition algorithm based on stacking denoising autoencoder and lightGBM. Sensors 19:947. https://doi.org/10.3390/s19040947
28. Gu F, Khoshelham K, Valaee S et al (2018) Locomotion activity recognition using stacked denoising autoencoders. IEEE Internet Things J 5:2085–2093. https://doi.org/10.1109/JIOT.2018.2823084
29. Mohd Noor MH, Ahmadon MA, Osman MK (2019) Activity Recognition using Deep Denoising Autoencoder. In: 2019 9th IEEE international conference on control system, computing and engineering (ICCSCE), pp 188–192
30. Gao L, Bourke AK, Nelson J (2014) Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems. Med Eng Phys 36:779–785. https://doi.org/10.1016/j.medengphy.2014.02.012
31. Banos O, Galvez J-M, Damas M et al (2014) Window size impact in human activity recognition. Sensors 14:6474–6499. https://doi.org/10.3390/s140406474
32. Fida B, Bernabucci I, Bibbo D et al (2015) Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer. Med Eng Phys 37:705–711. https://doi.org/10.1016/j.medengphy.2015.04.005
33. Noor MHM, Salcic Z, Wang KI-K (2017) Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. Pervasive Mob Comput 38:41–59. https://doi.org/10.1016/j.pmcj.2016.09.009
34. Vincent P, Larochelle H, Bengio Y, Manzagol P-A (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning. ACM, New York, NY, USA, pp 1096–1103
35. Ismail Fawaz H, Forestier G, Weber J et al (2019) Deep learning for time series classification: a review. Data Min Knowl Discov 33:917–963. https://doi.org/10.1007/s10618-019-00619-1
36. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
37. Goodfellow I, Pouget-Abadie J, Mirza M et al (2014) Generative Adversarial Nets. In: Ghahramani Z, Welling M, Cortes C et al (eds) Advances in neural information processing systems 27. Curran Associates, Inc., Red Hook, pp 2672–2680
38. Reyes-Ortiz J-L, Oneto L, Samà A et al (2016) Transition-aware human activity recognition using smartphones. Neurocomputing 171:754–767. https://doi.org/10.1016/j.neucom.2015.07.085

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