Neurocomputing 439 (2021) 271–280

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An ensemble of autonomous auto-encoders for human activity recognition

Kemilly Dearo Garcia [a,b,*], Cláudio Rebelo de Sá [a], Mannes Poel [a], Tiago Carvalho [a], João Mendes-Moreira [a], João M.P. Cardoso [c], André C.P.L.F. de Carvalho [b], Joost N. Kok [a]

[a] University of Twente, The Netherlands
[b] University of São Paulo, Brazil
[c] University of Porto, Portugal

ARTICLE INFO

Article history:
Received 27 September 2019
Revised 12 December 2019
Accepted 8 January 2020
Available online 26 January 2021

Keywords:
Human activity recognition
Ensemble of auto-encoders
Semi-supervised learning

ABSTRACT

Human Activity Recognition is focused on the use of sensing technology to classify human activities and to infer human behavior. While traditional machine learning approaches use hand-crafted features to train their models, recent advancements in neural networks allow for automatic feature extraction. Auto-encoders are a type of neural network that can learn complex representations of the data and are commonly used for anomaly detection. In this work we propose a novel multi-class algorithm which consists of an ensemble of auto-encoders where each auto-encoder is associated with a unique class. We compared the proposed approach with other state-of-the-art approaches in the context of human activity recognition. Experimental results show that ensembles of auto-encoders can be efficient, robust and competitive. Moreover, this modular classifier structure allows for more flexible models. For example, the extension of the number of classes, by the inclusion of new auto-encoders, without the necessity to retrain the whole model.

1. Introduction

Human Activity Recognition (HAR) is a research field focused on the use of sensing technology to classify human activities and to infer human behavior [1]. A HAR system can use data from different sources, like wearables, sensors from objects and cameras. These systems have been successfully applied for health and well-being [2], tracking and mobile security [3] and elderly care [4].

Most HAR machine learning approaches found in the literature, such as: decision trees [5], support vector machines [6] and *k*-Nearest Neighbor [4] rely on the use of heuristic hand-crafted feature extraction to train their models. That includes, for example, time-domain calculations, mean and standard deviation for each sensor signal and correlation (Pearson correlation) between axes for the 3D sensors.

In our previous work [7] we studied a semi-supervised ensemble, E*k*VN, which combined 3 different algorithms (*k*Nearest Neighbour, Very Fast Decision Tree and Naive Bayes). This method relies on heuristic hand-crafted feature extraction for HAR. The features were extracted from the raw data of different types of sensors: accelerometer, gyroscope and magnetometer sensors. We investigated the impact of some hyperparameters in the accuracy of E*k*VN. We found that the accuracy of E*k*VN is more sensitive to data from different users, to the window size and to the overlapping factor. We also found that the feature extraction process has a relatively high energy and time costs. This can have implications, for example in mobile applications, where the use of resources must be carefully managed in order to keep the application efficiently working for long periods of time.

An alternative to the manual extraction of features is the automatic feature extraction with neural networks [8]. One type of neural network commonly used as a powerful tool for discovery of features is the Auto-Encoder (AE). This type of neural network tries to learn two functions, an encoder, which maps the input to the hidden layers (the bottleneck), and a decoder, which maps the hidden layers to the output layer. In other words, an AE can learn compact representations of the input data in an unsupervised manner [9]. Therefore, the output of an auto-encoder is the reconstruction of its input.

In this work, an extension of [7], we propose a classification approach which is an Ensemble of AEs (EAE). In this EAE each AE is trained with data from one class.[1] Thus, in the context of HAR,

* Corresponding author.
E-mail address: k.dearogarcia@utwente.nl (K.D. Garcia).

[1] This type of ensemble might also be known as Mix of Experts [10].

https://doi.org/10.1016/j.neucom.2020.01.125
0925-2312/

each AE is associated with a label/activity. As new data arrives for classification, the reconstruction loss is calculated for each AE. The data is then classified with the label from the AE which obtained the lowest reconstruction loss. When used in online learning, the ensemble model can be updated with the user's data when the reconstruction loss drops below a given threshold. To the best of our knowledge there are no approaches that use AEs as an ensemble classifier.

We tested two variants of EAE in HAR data, an online and an offline one. Both variants learn from the same train data, however the first also learns incrementally when the loss increases more than an user-defined threshold. Experimental results show that the EAEs are efficient, robust and competitive with state-of-the-art approaches.

This paper is structured as follows. Section 2 presents the related work on machine learning for HAR. Section 3 describes the method proposed in this study. The results obtained are presented and discussed in Section 4. Finally, Section 5 summarizes the main conclusions and points out future work directions.

2. Related work

The main goal of HAR is to recognize human physical activities from sensing data. In this research area many approaches were presented in the last decade [11–14]. These approaches vary depending on the sensor technologies used to collect the data, the machine learning algorithm and the features created to train the model. In relation to extraction and selection of features, the models can be trained using hand-crafted feature extraction or automatic feature extraction.

The conventional approaches in HAR use hand-crafted feature extraction, which means that these approaches rely on human domain knowledge. Those features often include statistical information, such as: mean, variance, standard deviation, frequency and Pearson Correlation [15]. These approaches use traditional machine learning methods such as: SVM classifiers, *k*-Nearest Neighbour, decision tree, Naive Bayes classifiers, Random Forest [6,16,5,1]. Others focus on the combination of these machine learning approaches as ensembles in order to improve accuracy [7,1]. It is generally known that ensembles with bagging and boosting techniques can increase the performance of classifiers [17]. In most of these studies the improvements proposed are more focused in the tuning of hyperparamenters that are common in HAR (e.g. window size and overlapping factor) [18] and feature construction [1].

In contrast to that, Neural Networks methods (a.k.a Deep Learning) have the capacity to automatically learn relevant features from raw data without human domain knowledge [9]. Many different deep learning architectures have been proposed, such as Convolutional Neural Networks (CNN) [12,14,19], recurrent neural networks [13] and AEs [20–22].

Mostly used for computer vision, CNN models have also demonstrated to be effective in natural language processing [23], speech recognition [24] and text analysis [25]. In terms of HAR, CNNs have also been used to extract features from sensing data and to classification tasks [9]. Approaches for HAR based on CNNs can learn the correlation between nearby signals and be scale-invariant for different frequencies [9,19]. Some of these approaches process each dimension of a signal (e.g. a 3D accelerometer signal) as a channel. In other words, that means that to each channel is applied a 1D convolution. After that, the outputs from all channels are flattened to unified layers. Chen and Xue [26] used a CNN model with a modified convolution kernel to adapt to the characteristics of 3D signals. On the other hand, 2D convolutions can present better results compared to 1D convolutions. Ha and Choi [27] proposed 2D convolutions where CNNs were used with partial and full weight sharing structures to investigate the performance of different weight-sharing techniques. Weight-sharing is a technique used to incorporate invariance, to reduce complexity and to speed up the training process of CNNs [9]. To use 2D convolutions, some approaches resize the inputs from the signals as virtual 2D images [28]. To learn the dependencies between signals they applied a CNN using a 2D convolution kernel and a 2D pooling kernel. Following this idea Jiang and Yin [29] designed a more complex process to transform the signals into 2D image description and applied 2D convolution to extract features.

AEs are one family of neural networks which can learn a compact representation of the input signals. Stacked Auto-Encoder (SAE), for example, stack the learned features which can later be used to build a classification model [9]. Wang et al. [20] proposed a Continuous AE that converts high-dimensional continuous data to low-dimensional data in the encoding process. The features are extracted by AEs with multiple hidden layers. Gao et al. [21] proposed the combination of Stacking Denoising AE for feature extraction with LightGBM as the classifier. Ensemble of AEs can also be used for unsupervised outlier detection. For example, Chen et al. [30] proposed an ensemble of AEs randomly connected with different structures and connection densities, which reduces computational costs. The outliers are detected by computing the median of the AEs reconstruction error. In HAR, the features learned by Denoising Stacked AEs can be used by a random forest algorithm to build an ensemble classifier [31].

3. Methodology

In this section we start by describing the E*k*VN method with the hand-crafted feature extraction, presented in [7]. Afterwards, we describe the proposed method, the Ensemble of Auto-Encoders (EAE). Both methods are semi-supervised learning approaches. This means that they are incrementally updated after the data from a specific user is classified.

3.1. Ensemble of kVN

The E*k*VN is an ensemble model composed by three classifiers: *k*NN, Very Fast Decision Tree (VFDT) and Naive Bayes. The implementation of the ensemble classifier is the combination of Democratic Co-Learning and Tri-Training [32]. This method uses a vector of hand-crafted features as input, both in its training and test phase, as illustrated in Fig. 1.

The top pipeline in Fig. 1 shows the offline training using raw data extracted from different wearables and/or smartphone sensors. In the first step, *window segmentation* & *overlapping*, the raw data is stored in sliding windows and consecutive windows are overlapped. The window size (w) and overlap factor (ovl) are user defined values.

Data from sensors is usually susceptible to noise, especially accelerometer data [1]. Thus, the *preprocessing* step is important for calibration and filtering of the input data in order to reduce the noise. After that, a new instance is created containing the features that will be used to train the model, the *feature extraction* step. These features include time-domain calculations, specifically the mean, the standard deviation and the Pearson Correlation of each axis for the 3D sensors. Afterwards these instances are used to train (*training* step) one model from each one of the algorithms: *k*NN, VFDT and Naive Bayes. Then, they are combined as an ensemble of models.

In the online phase, new data is collected from a specific user. This data is preprocessed as described in the steps from the training phase: *window segmentation* & *overlapping*, *preprocessing* and *feature extraction*. Each new instance is classified by the ensemble,

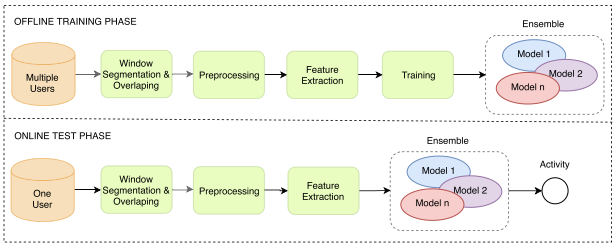

Fig. 1. Overview of the ensemble model, E*k*VN, for HAR.

which provides a confidence factor for the classification. The instances classified with high confidence, more than 99%, are used to update the model.

3.2. Ensemble of auto-encoders

A basic AE is a neural network model in which the output replicates the input, $y^i = x^i$ [20]. An AE consists of two parts, an Encoder and a Decoder, Fig. 2. The encoder learns to compress the inputs into a smaller number of encoded features, which is called the bottleneck. Given the encoded features, the decoder learns how to reconstruct the original input. Therefore, the output of an AE is an approximate reconstruction of the input [13].

In this work, we propose to use a set of AE, as an ensemble, for classification. The code is available on GitHub.[2] Fig. 3 illustrates the steps for training the Ensemble of Auto-Encoders (EAE) (offline phase) and how it can be used for classification (online phase). In the offline phase a batch of data from multiple users is used to train one AE per class. Thus, each AE learns a different activity.

In the online phase, as new data arrives, each AE tries to reconstruct the original input. Then the AE with the smallest reconstruction error, *minError*, is selected. The data is then classified with the label corresponding to the AE with the *minError*. During this online phase, each AE is updated whenever the reconstruction error falls below a user-defined threshold, *T*. By default, we define this threshold as *X* standard deviations of the training error. The threshold is a hyperparameter to set a high confidence factor, as in the method explained in Section 3.1. In both offline and online phases, the raw data is segmented according to a user defined window size (*w*) and an overlapping factor (*ovl*).

To illustrate how the EAE works, we present in Fig. 4 a simple example. The red line represents the real signal (used as the input data) and the blue line depicts the reconstructed signal by each AE. In this example, the model is composed by 6 AEs, where each was trained with data from one of the following activities: *Walking Downstairs*, *Jogging*, *Sitting*, *Standing*, *Walking Upstairs* and *Walking*. In Fig. 4 one can see that the AE which better reconstructs the signal is the *Sitting* AE. Therefore, the model classifies this activity as *Sitting*. Finally, if its error is below the defined threshold *T*, the AE is updated with this new signal.

4. Experiments

We conducted several experiments to compare the predictive performance of the EAE, with the E*k*VN and 5 other deep learning approaches. These methods are briefly described in Section 2 and as in [33] will be referred by the name of their author as: ChenXue [26], HaChoi [27], Haetal [28], JiangYin [29], Panwaretal [19]. The performance of all the methods was tested in 3 datasets commonly used in the literature of HAR.

[2] https://github.com/Keh/EAE.git.

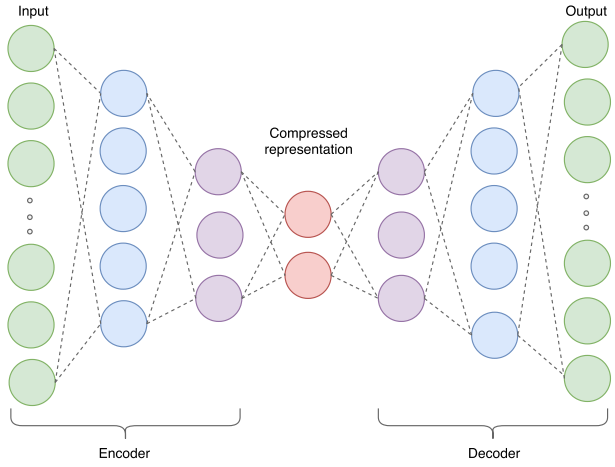

Fig. 2. An auto-encoder fully-connected structure with 3 hidden layers.

4.1. Datasets

In Table 1 we can see some statistics with a brief description of each dataset. Fig. 5 illustrates the frequency of activities that each dataset has. They all include standard activities, such us: *Walking*, *Jogging*/*Running*, *Standing*, *Sitting* and *Climbing Stairs*. The activities can be divided in Static, such us *Standing* and *Sitting*, and Dynamic, such us *Walking* and *Jogging*. The datasets MHealth and PAMAP2 also have more complex activities, such us house cleaning or sports. By complex activities we refer to activities that can be decomposed into others activities. For example, *Vacuum Cleaning* can be decomposed in: *Standing*, *Walking* and *Bending Forward*.

The *WISDM* dataset [34] contains sensor data from phone-based accelerometers.[3] The data was collected by an application installed on each user's phone. It has 1.098.209 records, of a 3-axis accelerometer sensor, from 29 users carrying a smart-phone placed on their front pants' pocket. In this dataset, there is no information about the age, gender or physical/behavior characteristics of the users. The data was collected at 20 Hz samples per second. The distribution of the classes can be seen in Fig. 5. The most common activities are: *Jogging* and *Walking*.

The *MHealth* dataset [2] has sensor data from 10 users performing 12 activities .[4] The data was collected from 3 devices with the following embedded sensors: a 3-axis accelerometer, a 3-axis gyroscope, a 3-axis magnetometer and an electrocardiogram sensor. These sensors were placed on different body locations, such as, chest, hand and ankle. There is also no personal information about the users. This dataset has 1.215.745 instances in total and has reasonably well balanced classes. The class with less data is *Jump front & Back*.

Finally, the *PAMAP2* dataset [35] is a public dataset of human physical activities.[5] The data was collected from 3 devices positioned in different body locations: hand, chest and ankle. Each device has three embedded sensors: a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer. This dataset contains 1.926.896 samples of raw sensor data from 9 different users and 18 activities. The authors divided these activities in: basic activities (*Walking* and *Running*), posture activities (*Lying* and *Standing*) and house cleaning (*Ironing* and *Vacuum Cleaning*). Also, part of the users performed optional activities, such us *Rope Jumping*.

[3] http://www.cis.fordham.edu/wisdm/dataset.php.
[4] http://archive.ics.uci.edu/ml/datasets/mhealth+dataset.
[5] http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring.

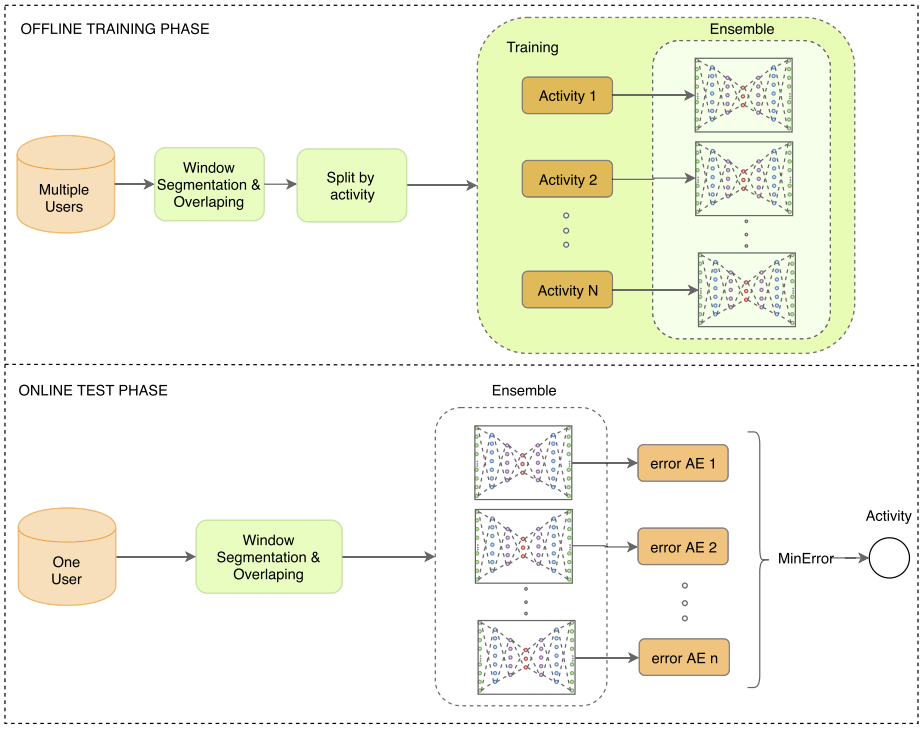

Fig. 3. Overview of the ensemble model, EAE, for HAR.

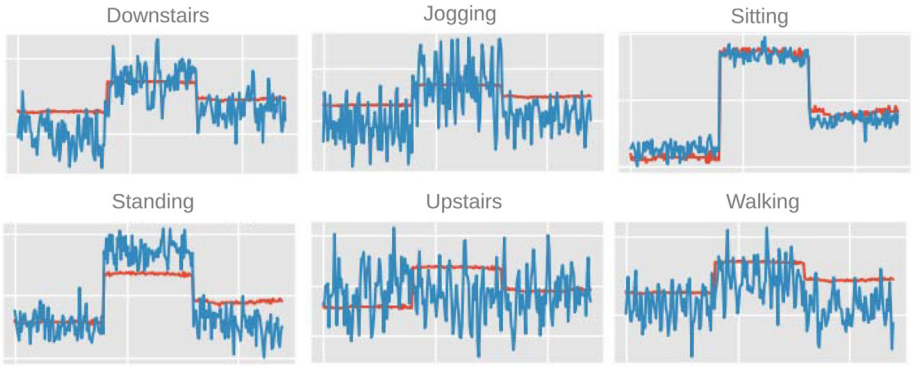

Fig. 4. The reconstruction of the signal by each AE. The color *red* represents the original signal and the color *blue* represents the reconstructed signal.

Table 1
Details of the 3 HAR datasets used in this work (A = accelerometer, G = gyroscope, M = magnetometer, C = electrocardiograph).

Dataset	#Users	S. Rate	#Activity	#Samples	Sensors	Body location
WISDM	36	20 Hz	6	1.098.209	A	Front pants' pocket
MHealth	10	50 Hz	12	1.215.745	A, G, M, C	Chest, Hand, Ankle
PAMAP2	9	100 Hz	18	1.926.896	A, G, M	Chest, Hand, Ankle

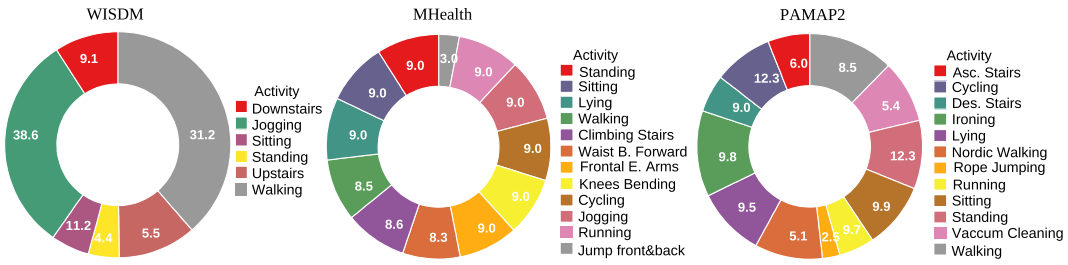

Fig. 5. Frequency per class of each dataset.

4.2. Experimental setup

We analyzed the performance of all the tested methods in terms of accuracy and computational cost. The latter was measured in seconds both in training and testing. For a fair comparison between the models, we only used the accelerometer data from each dataset. We trained the models with a fixed window size, w, of 160. Although other alternatives can be considered for the choice of the window size, for example, dynamic window size [36], it would require additional steps such as compression or concept drift detectors, which would increase the computational cost. Therefore, for simplicity, we used a fixed window $w = 160$. In practical terms, this represents 8.0 seconds for WISDM (20 Hz), 3.2 seconds for MHealth (50 Hz) and 1.6 seconds for PAMAP2 (100 Hz). Each consecutive window is overlapped with *ovl*, overlap factor, of 20% [7]. For the evaluation we used the leave-one-user-out approach.

We note that in the case of the proposed EAE the input is a vector with 480 entries. Which consists of the 3 components of the accelerometer sensor: x-acceleration, y-acceleration, and z-acceleration.

For the E*k*VN, we created the features: mean, standard deviation and Pearson Correlation. We also use the confidence factor of 99% for updating the model.

In the EAE method, each AE is composed of 8 hidden layers. The encoder has one input layer with 480 nodes, and 4 hidden layers with respectively, $200, 100, 80$ and 32 nodes. The decoder has the opposite structure: 4 hidden layers of $32, 80, 100$ and 200 nodes and with an output layer of 480. The first and the second layers of the AE have the *ReLU* activation function and the third and last layer a *linear* activation function. Each AE was trained for 250 epochs with a shuffled batch of size 256. The loss function used was the MAE and the optimizer was the adaptive moment estimation (Adam). In the online phase, the AE is updated when the *minError* (Section 3) is less $minError \leqslant threshold, T$, where $T = 0.01$.

To train the methods ChenXue, HaChoi, Haetal, JiangYin, Panwaretal, we used the same configuration proposed by the authors. The only difference is that we use the same window size $w = 160$, for a fair comparison with the other methods.

For each dataset, we analyse the mean and the dispersion of the accuracy per model. For that, in the first analysis, we are including all the experiments in which the data was divided by user. In the second analysis we focus in the accuracy of the models per user. Due to space constrains, we only present the results from the experiments with data collected from accelerometers placed in one body location, the hand or the pocket. Finally, we measure how the accuracy varies with the data collected from an accelerometer placed on different body locations. In this third analysis we present the average accuracy of the models per body location.

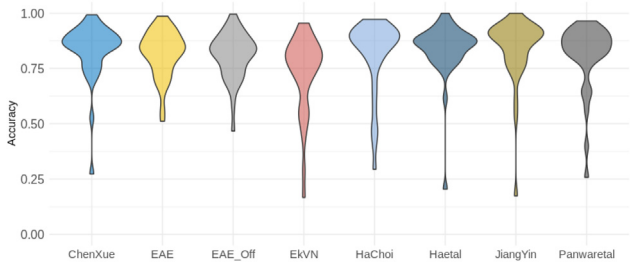

Fig. 6. Violin box-plot showing the dispersion of accuracy of the models in the dataset WISDM.

4.3. Results and discussion

In this section, we present and discuss the main results from the experiments. We note that in the experiments that includes the deep learning models, we present the results of our method with incremental learning, called EAE, and the model without incremental learning, called EAE_Off. We present both versions so we can analyze the improvement of online model update and also for a fair comparison with the other deep learning models that are not updated online.

4.3.1. WISDM dataset

Considering the WISDM dataset, we can observe a plot containing 8 violin box-plots representing the variation/dispersion of the accuracy per model Fig. 6. In this graph, each model is represented by a different color. Each box-plot has the results of all the experiments concerning the accuracy of each model per user. In Fig. 6 we notice that EAE has less dispersion in accuracy than the other models. The median of the accuracy is around 87% for EAE, while for E*k*VN it is around 80%. As for the other models, the median accuracy is around 87%, however their variance is larger than the variance of EAE. As for the lowest accuracy, it can reach in some cases, less than 25%.

We can see in Table 2 that the deep learning models have similar average accuracy, however the models Haetal and JiangYin show slightly better results. The average accuracy of EAE and E*k*VN models are 0.82 and 0.73, respectively. In terms of computational cost, we see that the model JiangYin took more time to train than the other methods. The EAE model has an average training time similar to the HaChoi, Haetal and Panwaretal models. The models E*k*VN model and ChenXue have the lowest training time. The ChenXue model has the simplest deep learning architecture, so it is reasonable that its time consumption is lower than the others. In terms of the testing computational cost, the EAE has the highest one. This can be due to the number of AEs and also the incremental learning step. Overall, the results show that both the EAE are learning meaningful representations of the activities in a reasonable time. However, the time for prediction is superior due to the number of AE models and its incremental learning.

In Fig. 7 we see the accuracy per user for the models EAE and the E*k*VN. The EAE obtained a higher accuracy in 78% of the users as compared with E*k*VN. One of the most striking differences is in user 30, where the accuracy of the EAE model is 71% while the accuracy of E*k*VN model was only 16.7%. As mentioned before, we do not have demographic information about the users, however we observe that the misclassification between *Walking* and *Jogging* was more evident in some users than others. Since the difference between the activities is in the intensity of the movement, it could have been useful to compare physical characteristics of the users with the classification.

When looking at the confusion matrix of the EAE (Table 3), we can observe that the classes with higher misclassification are

Table 2
Average accuracy (Acc) and time (train/test) for the models in the WISDM dataset.

	ChenXue	HaChoi	Haetal	JiangYin	Panwaretal	EAE	EAE_Off	EkVN
Acc	0.83	0.81	**0.84**	**0.84**	0.81	0.82	0.81	0.73
Time (s)	**65**/**0.1**	104/0.1	196/0.1	430/0.1	109/0.1	172/20.0	172/14.0	50/0.2

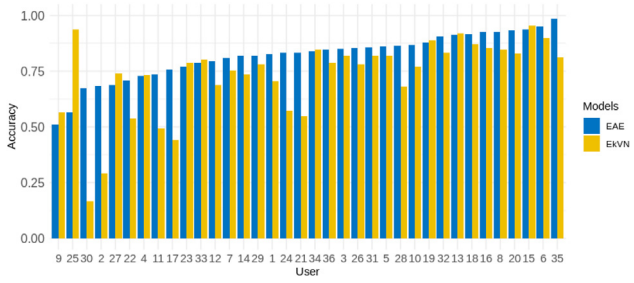

Fig. 7. Accuracy per user for the EAE and the E*k*VN models for the WISDM dataset considering the body location *Pocket*.

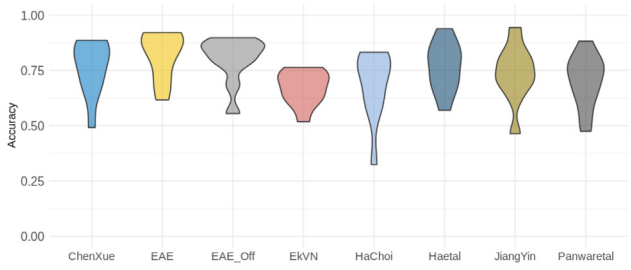

Fig. 8. Violin box-plot showing the dispersion of accuracy of the models in the dataset MHealth.

Downstairs and *Upstairs*. They are often misclassified with each other or with *Walking*. One difference between the classes *Downstairs* and *Upstairs* is the orientation of the activity: one is descending stairs and the other is ascending stairs. This concept might be hard to learn only from accelerometer data, since this sensor does not capture the orientation of the movement. On top of that, we also notice that the AEs *Downstairs* and *Upstairs* were trained with less data than others classes (Fig. 5) which makes it even more difficult for the models to learn them.

4.3.2. MHealth dataset

In Fig. 8, we can observe the dispersion of accuracy obtained by the models in the MHealth dataset. The median of accuracy of the EAE model is above 90%. We note that the EAE has less variance than EAE_Off, meaning that the incremental learning reduces variance. The models ChenXue, HaChoi and JiangYin had higher variance than the other models. Although the lowest variance is E*k*VN, its median accuracy of 75%.

For this experiment we consider only data from the body location *hand*. In terms of average accuracy and time consumption, we can see in Table 4 that both the EAE and the EAE_Off are competitive results with the deep learning models. However, because of the incremental learning of the EAE it obtained an even higher average accuracy than other models. On the other hand, the only model that uses hand-crafted features, E*k*VN, had the lowest accuracy.

In terms of time consumption, one more, the models with simpler architectures are faster to train (HaChoi and ChenXue). The EAE takes more time in the prediction phase, specially because this phase includes the incremental learning of the model. Considering that this is an ensemble, the amount of models influences on the time consumption of its testing phase.

When comparing the accuracy per user of the models EAE and E*k*VN (Fig. 9) we can observe that the EAE was better for all individuals. This shows, once again, that the proposed method can learn meaningful representations of the activities.

By looking at the confusion matrix of the EAE model (Table 5) we see that the class *Stairs* has an average accuracy of 93%. This class has data from *Downstairs* and *Upstairs* combined. This shows that the model can learn better from the classes which are independent of the orientation. The class *Running* was more misclassified as *Jogging* than the other way around, which might be related to the pace that each individual takes to perform these activities.

4.3.3. PAMAP2 dataset

For the PAMAP2 dataset, we see in Fig. 10 a high dispersion in the accuracy of the models, specially for ChenXue, HaChoi and Panwaretal. Although E*k*VN also has a high dispersion of the accuracy, it is the model with the highest median accuracy, around 70%. The model EAE has the median of accuracy slightly above 60%, presenting a small improvement compared with the offline variant, EAE_Off.

The maximum accuracy reached by EAE is 0.91, which is the highest of the deep learning methods. We also observed that the minimum accuracy of the EAE is always the highest in all the datasets tested. In this case, the lower value is 0.44 which is the same as for E*k*VN.

In Table 6 we see that the average accuracy of the E*k*VN the highest, meaning that traditional models can achieve better performance in some datasets. Haetal is the deep learning model with the highest average accuracy. All the others, JiangYin, Panwaretal, ChenXue, EAE, EAE_Off and HaChoi obtained very similar average accuracy.

In terms of time consumption, we see that ChenXue has a faster training time, however the JiangYin and Panwaretal are faster for testing. We notice that, although EAE is an ensemble of 12 AEs the time of training is not higher that some other deep learning models (e.g. Panwaretal). However the EAE model is the slowest in testing time. The time performance depends on the complexity of the models, the amount of models and the amount of data. Therefore it is natural that EAE shows a higher consumption time.

Considering the results per user of the models EAE and E*k*VN (Fig. 11), we see that the accuracy of E*k*VN was slightly higher for all individuals. However, in this dataset, the analysis per user is not an easy task because the users did not perform the activities in equal proportion. For example, the user 9 only performed the activity *Jumping*. This is reflected in Fig. 5 where we can see that there is less data for some classes. This less amount of data has obvious implications in the deep learning methods which are

Table 3
Average confusion matrix for the EAE model in the WISDM dataset considering the body location *Pocket*. The columns represent the ground truth and rows represent the predicted.

	Downstairs	Jogging	Sitting	Standing	Upstairs	Walking
Downstairs	**0.40**	0.06	0.01	0.00	0.22	0.30
Jogging	0.01	**0.96**	0.00	0.00	0.01	0.02
Sitting	0.02	0.03	**0.91**	0.04	0.01	0.00
Standing	0.01	0.00	0.04	**0.95**	0.00	0.00
Upstairs	0.06	0.11	0.00	0.00	**0.58**	0.24
Walking	0.09	0.00	0.00	0.00	0.04	**0.87**

Table 4
Average accuracy (Acc) and time (train/test) for the models in the MHealth dataset.

	ChenXue	HaChoi	Haetal	JiangYin	Panwaretal	EAE	EAE_Off	EkVN
Acc	0.76	0.69	0.77	0.74	0.70	**0.82**	0.75	0.67
Time (s)	65.1/0.1	**49.4/0.1**	385.1/0.4	83.3/0.1	197.0/0.1	209.9/36	209.9/31	79.3/1.1

known to require more data than classical approaches. This is more evident for the activities *Ascending Stairs*, *Nordic Walking* and *Rope Jumping*. This dataset also have activities like *Ironing* and *Vacuum Cleaning* which are a mix of activities, such us, *Walking* and *Standing*.

The average Confusion Matrix of PAMAP2 (Table 7) shows that the misclassification occurs between classes that are not related. For example, *Descending Stairs* and *Ironing*. From this, we can conclude that the EAE model did not learn the activities as good as in the other datasets. The reason for that might be because this dataset was collected with a frequency of 100 Hz (see Table 1). Because of the high frequency, the window size of 480 data points (160 data points per accelerometer axis) represents only 1.6 seconds for each activity, which is not sufficient to learn meaningful representations of each activity. Thus, a bigger window should have been used for this dataset.

4.4. Accuracy per body location

In Table 8, we present the average accuracy per body location of the models EAE and E*k*VN, considering WISDM, MHealth and PAMAP2 dataset. For the WIDSM dataset, that only have the body location Front Pocket, the average accuracy of EAE is higher than the E*k*VN model. In terms of the sensors on different body locations for the dataset MHealth, the average accuracy of the EAE for each position is higher than the E*k*VN. Moreover, we can see that the accuracy of the EAE models is practically the same across the different body locations, while the E*k*VN varies. Additionally, as expected, the combination of all the sensors placed on different body locations (HCA) improved the results of both models. For the dataset PAMAP2 we see that the average accuracy of the E*k*VN for each body location is higher than the EAE model. This is specially evident in HCA. However the EAE model has a lower variance, since all values are around 60.0%.

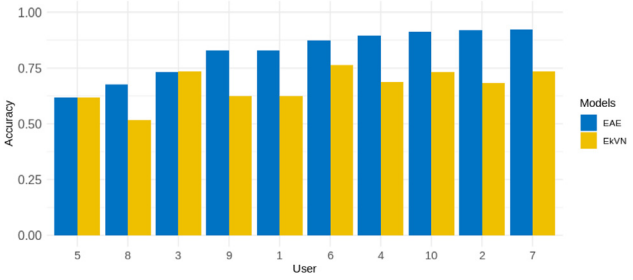

Fig. 9. Accuracy per user for the EAE and the E*k*VN models in the MHealth datasets, considering the body location *Hand*.

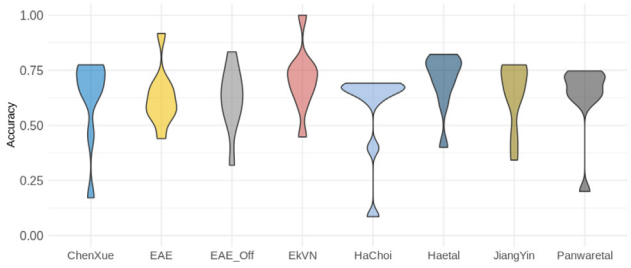

Fig. 10. Violin box-plot showing the dispersion of accuracy of the models in the dataset PAMAP2.

4.5. Aggregation of classes

One advantage of the EAE structure is the possibility of aggregating classes in different hierarchies in a simple manner. We can combine the AEs that represent similar classes and consider super-classes.

Considering the dataset PAMAP2, for example, we can aggregate its classes into the following super-classes: *House Cleaning* (which includes *Ironing* and *Vacuum Cleaning*), *Dynamic Positions* (*Ascending/Descending Stairs* and *Walking*), *Static Positions* (*Lying*, *Sitting* and *Standing*) and *Sports* (*Cycling*, *Nordic Walking*, *Rope Jumping* and *Running*). In this experiment, we still use the 12 AEs, however we consider as a true positive, when the classification is correct, any class belonging to the super-class (Table 9). The average accuracy was 74.1%, which is higher than the 60.0% showed in Table 7. This shows that AE from similar activities obtain smaller errors. In particular, the misclassification between the *Dynamic Positions* and *Static Positions* is quite low.

Table 5
Average confusion matrix for EAE model for MHealth dataset considering only the body location *Hand*. The columns represent the ground truth and rows represent the predicted.

	1	2	3	4	5	6	7	8	9	10	11	12
1 - Stand	**0.89**	0.04	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2 - Sit	0.11	**0.45**	0.22	0.11	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00
3 - Lay	0.00	0.11	**0.78**	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00
4 - Walk	0.01	0.00	0.00	**0.95**	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5 - Stairs	0.00	0.01	0.00	0.02	**0.93**	0.02	0.01	0.02	0.00	0.00	0.00	0.00
6 - Waist Bend	0.01	0.01	0.00	0.01	0.02	**0.74**	0.00	0.22	0.00	0.00	0.00	0.00
7 - Elevation	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00
8 - Knees Bend	0.00	0.01	0.00	0.04	0.06	0.23	0.00	**0.66**	0.00	0.00	0.00	0.00
9 - Cycle	0.00	0.19	0.00	0.00	0.00	0.00	0.00	0.00	**0.81**	0.00	0.00	0.00
10 - Jog	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.92**	0.08	0.00
11 - Run	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.27	**0.73**	0.00
12 - Jump	0.00	0.00	0.02	0.06	0.02	0.00	0.00	0.00	0.00	0.02	0.00	**0.88**

Table 6
Average accuracy (Acc) and time (train/test) for the models in the PAMAP2 dataset.

	ChenXue	HaChoi	Haetal	JiangYin	Panwaretal	EAE	EAE_Off	EkVN
Acc	0.63	0.57	0.70	0.64	0.63	0.63	0.62	**0.71**
Time (s)	**79.3**/0.4	654.1/**0.3**	563.7/0.5	347.2/**0.3**	960.1/**0.3**	249.4/38.2	249.4/31	130.4/2.0

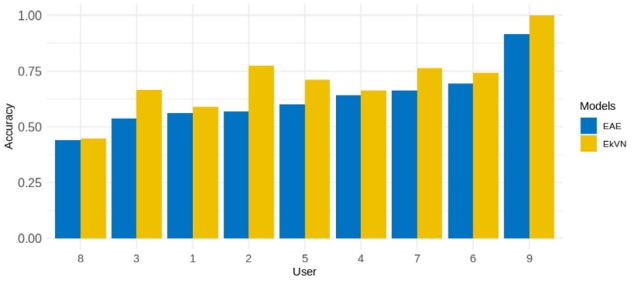

Fig. 11. Accuracy per user for the EAE and the E*k*VN models in the PAMAP2 datasets, considering the body location *Hand*.

5. Conclusion

In this paper we proposed a new classification algorithm which we refer as Ensemble of Auto-Encoders (EAE). It uses a set of AEs where each is trained to reconstruct the sensor measurements from one unique class. This set of AE is then used as an ensemble for classification by predicting the class which corresponds to the AE with the lowest reconstruction error. We tested two variants of the EAE (one with online learning and other without) in HAR datasets and compared them with other methods. One was an ensemble of traditional approaches, E*k*VN, and the remaining are state-of-the-art deep learning approaches.

Experimental results show that the proposed EAE is competitive with existing methods found in HAR literature. We observed that the minimum accuracy of the EAE is always the highest in all the datasets tested. From this we can conclude that the EAE is more robust to data from different users, which is also supported by the low variance in accuracy.

We note that the presented results were obtained from models trained with accelerometer data only, which is usually a more challenging classification task. Moreover, a simple and unique architecture was used for all AE in all datasets without hyperparameter tuning.

The modular structure of the EAE proposed in this work has the advantage of making the model easily adapted. First of all, in the case of online learning, only the AEs corresponding to the most frequent activities are updated which can save computation time. In

Table 7
Average confusion matrix for EAE model for PAMAP2 dataset considering only the body location *Hand*. The columns represent the ground truth and rows represent the predicted.

	1	2	3	4	5	6	7	8	9	10	11	12
1 – Asc. Stairs	**0.55**	0.00	0.10	0.05	0.01	0.01	0.00	0.00	0.03	0.03	0.09	0.13
2 – Cycle	0.01	**0.57**	0.01	0.12	0.02	0.00	0.00	0.00	0.13	0.08	0.05	0.01
3 – Des. Stairs	0.16	0.00	**0.42**	0.21	0.01	0.00	0.00	0.00	0.03	0.05	0.06	0.05
4 – Ironing	0.02	0.01	0.04	**0.60**	0.02	0.01	0.00	0.00	0.14	0.07	0.09	0.01
5 – Lay	0.01	0.00	0.01	0.06	**0.60**	0.00	0.00	0.00	0.28	0.03	0.01	0.00
6 – Nord. Walk	0.16	0.00	0.02	0.08	0.00	**0.40**	0.01	0.00	0.09	0.01	0.21	0.02
7 – Jump	0.01	0.00	0.01	0.09	0.00	0.01	**0.60**	0.01	0.07	0.02	0.19	0.00
8 – Run	0.01	0.00	0.00	0.02	0.01	0.00	0.00	**0.76**	0.01	0.02	0.17	0.00
9 – Sit	0.02	0.00	0.01	0.09	0.03	0.01	0.00	0.00	**0.73**	0.10	0.02	0.00
10 – Stand	0.06	0.00	0.03	0.06	0.07	0.00	0.00	0.00	0.12	**0.61**	0.03	0.01
11 – Vac. Clean	0.10	0.00	0.02	0.24	0.01	0.01	0.00	0.00	0.03	0.03	**0.52**	0.03
12 – Walk	0.21	0.00	0.11	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.06	**0.59**

Table 8
Average accuracy for the EAE and the E*k*VN models for each dataset separated by body location (HCA = Hand, Chest, Ankle).

	Hand		Chest		Ankle		HCA		F.Pocket	
	EAE	EkVN	EAE	EkVN	EAE	EkVN	EAE	EkVN	EAE	EkVN
WISDM	–	–	–	–	–	–	–	–	80.8	73.1
MHealth	82.0	67.2	82.8	74.4	82.0	69.2	94.8	83.4	–	–
PAMAP2	60.0	70.7	60.0	60.8	59.0	70.7	56.0	80.8	–	–

Table 9
Average confusion matrix for EAE model for PAMAP2 dataset considering the aggregation of AEs.

	Dynamic positions	Static positions	Sports	House cleaning
Dynamic Positions	**0.82**	0.05	0.00	0.13
Static Positions	0.05	**0.86**	0.01	0.09
Sports	0.09	0.13	**0.56**	0.23
House Cleaning	0.10	0.16	0.02	**0.72**

this way it is not necessary to retrain the whole model, as it would be necessary for most machine learning models. Therefore the EAE can specialize in the most performed (or preferred) activities of each user. Moreover this modular structure has also the advantage for the inclusion of new activities when is needed. For that, it is only necessary to add more AE and train each one with each new class. Likewise, it could be similarly adapted to forget activities, by simply removing the respective AE from the ensemble. Finally, another advantage of the EAE is that each AE can have its own architecture and even use different types of layers, such as Recurrent or Convolutional.

In terms of time consumption we see that models with more complex architectures are slower to train than simpler ones. In that sense the EAE, even though it has multiples models, has a similar time consumption to other deep learning models. However, since the concept of EAE can use many different architectures of the AEs, the time consumption can be reduced with different ones. Moreover, in terms of test/prediction, the time consumption represented in the results consider all the instances used in each experiment. Which means that the prediction of each instance took less than half a second.

As future work we intend to combine AE with different architectures in the same ensemble.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Dobbins, R. Rawassizadeh, E. Momeni, Detecting physical activity within lifelogs towards preventing obesity and aiding ambient assisted living, Neurocomputing 230 (2017) 110–132.

[2] O. Baños, R. García, J.A.H. Terriza, M. Damas, H. Pomares, I.R. Ruiz, A. Saez, C. Villalonga, mhealthdroid: A novel framework for agile development of mobile health applications, in: Ambient Assisted Living and Daily Activities – 6th International Work-Conference, IWAAL 2014, Belfast, UK, December 2–5, 2014. Proceedings, 2014, pp. 91–98. .

[3] S. Spinsante, A. Angelici, J. Lundström, M. Espinilla, I. Cleland, C.D. Nugent, A mobile application for easy design and testing of algorithms to monitor physical activity in the workplace, Mobile Inf. Syst. 2016 (2016) 5126816:1–5126816:17.

[4] S. Yao, S. Hu, Y. Zhao, A. Zhang, T.F. Abdelzaher, Deepsense: A unified deep learning framework for time-series mobile sensing data processing, in: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3–7, 2017, 2017, pp. 351–360. .

[5] O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors, IEEE Commun. Surveys Tutorials 15 (3) (2013) 1192–1209.

[6] A. Mannini, S.S. Intille, M. Rosenberger, A.M. Sabatini, W. Haskell, Activity recognition using a single accelerometer placed at the wrist or ankle, Med. Sci. Sports Exercise 45 (11) (2013) 2193.

[7] K.D. Garcia, T. Carvalho, J. Mendes-Moreira, J.M.P. Cardoso, A.C.P.L.F. de Carvalho, A study on hyperparameter configuration for human activity recognition, in: 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019) – Seville, Spain, May 13–15, 2019, Proceedings, 2019, pp. 47–56. .

[8] T. Plötz, N.Y. Hammerla, P. Olivier, Feature learning for activity recognition in ubiquitous computing, in: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 1729–1734. .

[9] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: a survey, Pattern Recogn. Lett. 119 (2019) 3–11.

[10] A.V. Makkuva, P. Viswanath, S. Kannan, S. Oh, Breaking the gridlock in mixture-of-experts: consistent and efficient algorithms, in: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA, 2019, pp. 4304–4313.

[11] A.H. Niazi, D. Yazdansepas, J.L. Gay, F.W. Maier, L. Ramaswamy, K. Rasheed, M. P. Buman, Statistical analysis of window sizes and sampling rates in human activity recognition, in: Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017) – Volume 5: HEALTHINF, Porto, Portugal, February 21–23, 2017, 2017, pp. 319–325..

[12] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J.L. Zhao, Exploiting multi-channels deep convolutional neural networks for multivariate time series classification, Front. Comput. Sci. 10 (1) (2016) 96–112.

[13] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, C.J. Spanos, Deepsense: device-free human activity recognition via autoencoder long-term recurrent convolutional network, in: 2018 IEEE International Conference on Communications, ICC 2018, Kansas City, MO, USA, May 20–24, 2018, 2018, pp. 1–6. .

[14] M.S. Seyfioglu, A.M. Özbayoglu, S.Z. Gurbuz, Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities, IEEE Trans. Aerosp. Electron. Syst. 54 (4) (2018) 1709–1723.

[15] D. Figo, P.C. Diniz, D.R. Ferreira, J.M.P. Cardoso, Preprocessing techniques for context recognition from accelerometer data, Pers. Ubiquit. Comput. 14 (7) (2010) 645–662.

[16] L. Bedogni, M. Di Felice, L. Bononi, By train or by car? Detecting the user's motion type through smartphone sensors data, in: 2012 IFIP Wireless Days, IEEE, 2012, pp. 1–6..

[17] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15. .

[18] O. Baños, J.M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, Sensors 14 (4) (2014) 6474–6499.

[19] M. Panwar, S.R. Dyuthi, K.C. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, G.R. Naik, CNN based approach for activity recognition using a wrist-worn accelerometer, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju Island, South Korea, July 11–15, 2017, 2017, pp. 2438–2441. doi: 10.1109/EMBC.2017.8037349. .

[20] L. Wang, Recognition of human activities using continuous autoencoders with wearable sensors, Sensors 16 (2) (2016) 189.

[21] X. Gao, H. Luo, Q. Wang, F. Zhao, L. Ye, Y. Zhang, A human activity recognition algorithm based on stacking denoising autoencoder and lightgbm, Sensors 19 (4) (2019) 947.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.

[23] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1746–1751. https://www.aclweb.org/anthology/D14-1181/. .

[24] O. Abdel-Hamid, A. Mohamed, H. Jiang, G. Penn, Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25–30, 2012, 2012, pp. 4277–4280. doi:10.1109/ICASSP.2012.6288864. https://doi.org/10.1109/ICASSP.2012.6288864 .

[25] C. dos Santos, M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 69–78. https://www.aclweb.org/anthology/C14-1008.

[26] A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer. doi:10.1109/SMC.2015.263. doi: 10.1109/SMC.2015.263. .

[27] Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. doi: 10.1109/IJCNN.2016.7727224. .

[28] Multi-modal Convolutional Neural Networks for Activity Recognition. doi: 10.1109/SMC.2015.525. .

[29] W. Jiang, Z. Yin, Human activity recognition using wearable sensors by deep convolutional neural networks, in: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26–30, 2015, 2015, pp. 1307–1310..

[30] J. Chen, S. Sathe, C.C. Aggarwal, D.S. Turaga, Outlier detection with autoencoder ensembles, in: Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27–29, 2017, 2017, pp. 90–98. doi:10.1137/1.9781611974973.11. .

[31] S. Thomas, M. Bourobou, J. Li, Ensemble of deep autoencoder classifiers for activity recognition based on sensor modalities in smart homes, in: Data Science – 4th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2018, Zhengzhou, China, September 21–23, 2018, Proceedings, Part II, 2018, pp. 273–295. doi: 10.1007/978-981-13-2206-8_24. .

[32] Z. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, IEEE Trans. Knowl. Data Eng. 17 (11) (2005) 1529–1541.

[33] A. Jordao, A.C.N. Jr., J.S. de Souza, W.R. Schwartz, Human activity recognition based on wearable sensor data: a standardization of the state-of-the-art, CoRR abs/1806.05226. arXiv:1806.05226. http://arxiv.org/abs/1806.05226 .

[34] J.R. Kwapisz, G.M. Weiss, S. Moore, Activity recognition using cell phone accelerometers, SIGKDD Explor. 12 (2) (2010) 74–82.

[35] A. Reiss, D. Stricker, Creating and benchmarking a new dataset for physical activity monitoring, in: The 5th International Conference on Pervasive Technologies Related to Assistive Environments, PETRA 2012, Heraklion, Crete, Greece, June 6–9, 2012, 2012, p. 40..

[36] C. Ma, W. Li, J. Cao, J. Du, Q. Li, R. Gravina, Adaptive sliding window based activity recognition for assisted livings, Inf. Fusion 53 (2020) 55–65, https://doi.org/10.1016/j.inffus.2019.06.013.

Kemilly Dearo Garcia is a PhD candidate in the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, The Netherlands. My main research topics are machine learning, concept-drift, data mining and learning from sensors data. My main application areas are currently in Health and Sports.

Cláudio Rebelo de Sá is a senior data scientist working at the University of Twente. I have a PhD in Computer Science from the University of Leiden. My main areas of expertise are Data Mining, Neural Network, Deep Learning and Computer Vision.

Mannes Poel, Assistant professor at the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, The Netherlands. My main research areas are Data Science, Machine Learning and Brain Computer Interfacing.

Tiago Carvalho is a PhD candidate in INESC TEC and Faculty of Engineering of the University of Porto (FEUP), at the Department of Informatics Engineering (DEI). I work with compiler-related topics such as domain-specific languages and compiler optimizations.

João Mendes-Moreira, Assistant professor in the Department of Informatics Engineering at the Faculty of Engineering from the University of Porto (FEUP), Portugal. My main areas of expertise are: predictive machine learning with special interest on ensemble learning, pocket data mining, intelligent transportation systems, and decision support systems.

João M.P. Cardoso, Full Professor at the Department of Informatics Engineering, Faculty of Engineering of the Univ. of Porto, Porto, Portugal and a research member of INESC TEC. My research interests include: Compiler Techniques, Reconfigurable Computing Platforms and Tools, and Design Automation for Embedded Systems.

André C.P.F.L. de Carvalho, Full Professor and Deputy Dean of the Institute of Mathematics and Computer Sciences, University of São Paulo, Brazil. My main research interests are data mining, data science and machine learning.

Joost N. Kok, Full Professor and Dean of the Faculty of Electrical Engineering, Mathematics and Computer Science at University of Twente, The Netherlands. My research is concentrated around data. My main research themes are data- and model-management, data mining, bioinformatics and algorithms. My main application areas are currently in Health, Sports and Energy.