Neural Computing and Applications (2020) 32:15673–15686
https://doi.org/10.1007/s00521-018-3437-x

S.I. : ADVANCES IN BIO-INSPIRED INTELLIGENT SYSTEMS

Comparison of offline and real-time human activity recognition results using machine learning techniques

Jozsef Suto[1] · Stefan Oniga[1,2] · Claudiu Lung[2] · Ioan Orha[2]

Received: 16 December 2017 / Accepted: 16 March 2018 / Published online: 20 March 2018

Abstract

Today's human activity recognition is an important part of healthcare and ambient-assisted living where accelerometer and gyroscope sensors provide the raw data about physical activities and functional abilities of an observed person. Previous studies have shown that activity recognition can be seen as a machine learning chain with its particular data preprocessing technique. In recent past, several scientists measured rather high recognition accuracies on public databases or in laboratory environment but their solutions have not been tested in real environment. The goal of this paper is to examine the efficiency of previously used machine learning methods in real time by an Android-based, self-learning, activity recognition application which has been designed especially to this study according to the latest theoretical results (with the most relevant feature extraction and machine learning algorithms). Before real-time tests, we investigated the design considerations and application possibilities of different shallow and deep methods. The final outcome shows recognition rate difference between the "online" and "offline" cases. In the article we present some reasons for the difference and their possible solutions.

Keywords Activity recognition · Android application · Feature extraction · Machine learning

1 Introduction

Physical activity is an important component of a healthy lifestyle. The goal of human activity recognition is to determine the activity level of someone. Monitoring and recognizing daily activities of a people can help in the evaluation and prediction of his/her health status. An interesting study [1] has shown that physically active people who live a healthier lifestyle have lower rates of diseases. Moreover, the rapidly growing rate of elderly population greatly influences the development of healthcare services. Based on these new challenges, many different approaches have been suggested by researchers for the recognition of physical activities in different application areas such as in ambient-assisted living and healthcare [2, 3]. In the past few decades, researchers tried different data acquisition approaches for human activity recognition (HAR). The two major techniques are based on computer vision and wearable sensor networks. Due to the disadvantages and limitations of camera-based methods (privacy issue, background change, lighting conditions, special environment, etc.), wearable sensors and their networks have received higher attention. The miniaturized sensor technology has made it possible for a person to wear data acquisition devices continuously on predetermined body segments. It motivated the researchers, research groups and companies to develop their own data acquisition devices with different kinds of sensors and controllers for HAR purposes [4–6]. According to the sensor technology development, the importance and popularity of HAR have notably increased in the past decade.

Today's smartphones also can be used as a complete HAR system without any additional hardware components. Already several researchers used phones for HAR [7]. Most

✉ Jozsef Suto
suto.jozsef@inf.unideb.hu

[1] Department of Informatics Systems and Networks, Faculty of Informatics, University of Debrecen, 26 Kassai Road, Debrecen 4028, Hungary

[2] Department of Electronic and Computer Engineering, Technical University of Cluj-Napoca, North University Centre of Baia Mare, 62A Str. Dr. Victor Babes, Baia Mare 43008, Romania

of them used the phone as a data acquisition device, and the evaluation happened offline by mathematical or data mining tools such as Weka, Python and MATLAB [8, 9]. Nowadays, almost everyone has a smartphone which is well equipped with fast processor(s), plenty memory, built-in sensors and powerful battery. Therefore, they are providing new opportunities in the HAR research. They have some advantages unlike special purpose data acquisition devices. For instance, smartphones are providing high-level programming environment with different visualization, data storage and communication capabilities. According to the previous reasons, in this study a smartphone acts in the sensor role and an Android-based, self-developed application performs the complete classification process which has been designed especially to this study according to the latest theoretical results.

Even though many data capture devices and different kinds of algorithms exist, activity classification is not an easy task. In order to the recognition be efficient, researchers applied stable and robust machine learning (ML) techniques that can handle noisy data. For example, Yang et al. [10], Khan et al. [11] and Oniga and Suto [12] used feed-forward artificial neural networks to the classification and measured 95 and 97.9 and 99% recognition rates, respectively. Preece et al. [13] and Duarte et al. [14] reached 95 and 97.8% accuracy with the *k*-nearest neighbour method, while Maurer et al. [4] and Gao et al. [15] measured similarly good rates (92.8, 96.4%) with decision trees. After the appearance of deep learning, some scientists from the HAR community turned towards deeper ML algorithms such as convolutional neural networks. They claimed that convolutional network is a better choice for HAR because it does not require feature extraction as shallow techniques. For instance, Sheng et al. [16] and Yiang and Yin [17] measured more than 95% recognition rates on public databases with this method. Consequently, the current state of the art proposes numerous ML solutions to HAR but we do not have any useful information about their usability in a real-time application. Therefore, the aim of this study is to investigate the efficiency and reliability of previous, promising offline results in real environment. To the best of our knowledge, it is the first work which compares offline and online results in this manner.

2 Methodology

Most works in HAR follow a general activity recognition chain. It contains data acquisition from sensor(s), data segmentation, feature extraction (sometimes feature selection), classifier training and classification. In this section, we will thoroughly describe each of these steps. Beyond the type of data capture devices and classifier algorithms, other questions also exist in this research field. The following subsections give an overview about the questions and their latest solutions which have been utilized in this study.

2.1 Number of sensors and sensor placement

Several researchers tried to find the most suitable body position for sensor placement which provides the best recognition accuracy. One part of researches investigated the usage of single sensor, placed on a specific body location. Godfrey et al. [18] used a single chest-mounted sensor for their study. Ayu et al. [9] worked with a single smartphone for data acquisition (on hand and in pocket), while in the work of Yang et al. [10] a single three-axis accelerometer was placed to the wrist of the observed person. Other part of studies applied multiple sensors on different body positions. Yang et al. [19] collected the data from five sensor nodes placed on the left and right ankle, left and right hand and hip. The authors of [13] worked with two sensors and placed them on the ankle and thigh. Our sensor (the smartphone) placement choice was motivated by several previous works. Gao et al. [15] presented that the recognition difference between multi-sensor and single-sensor systems is rather small. Ertugrul et al. [20] and Oniga and Suto [21] demonstrated that a single sensor with a three-axis accelerometer and gyroscope is enough for good daily activity recognition. Finally, Preece et al. [13] showed that the ankle is the most suitable place for single sensor. According to their results, in this study the sensor has been attached to the right ankle of the volunteers with a holder (Fig. 1). By this fixed vertical placement of the phone, we avoided the variable orientation problem which is the main disadvantage of a general smartphone-

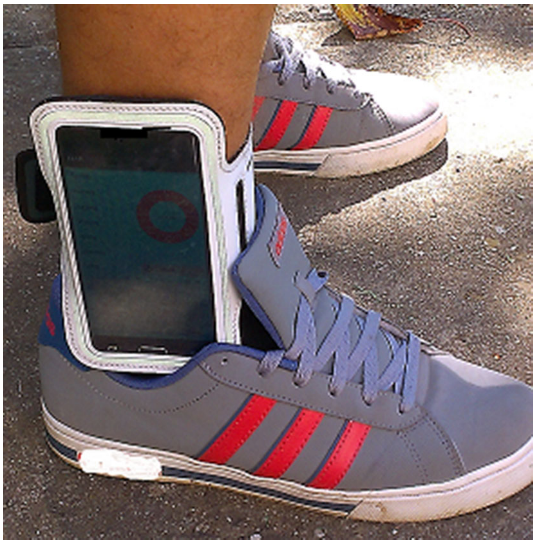


Fig. 1 Placement of the phone on the right ankle

based HAR systems. Another disadvantage of a phone against a special purpose device is its size. Such a size difference can be seen in Fig. 2. However, the size of the phone does not affect the data acquisition and the final outcome of this study.

2.2 Sampling

The sampling frequency is an important parameter in all signal processing applications because it greatly influences the power requirement, computational load and the performance. In some studies, the raw sensor output was oversampled. For instance, the sampling rate in the work of Yang et al. [10] was 100 Hz. However, such a high rate is unnecessary because the main frequency components of body movement are less than 10 Hz during daily activities [5, 22]. Gao et al. [15] examined the relation between sampling frequency and recognition rate, and they proposed 20 Hz sampling frequency to multi-sensor systems and a little higher rate to single-sensor systems. Maurer et al. [4] also showed that there is no significant improvement in recognition rate above 20 Hz. Other authors similarly used exactly or approximately 20 Hz sampling rate to the data acquisition such as Khan et al. [11] and Yang et al. [19]. According to these facts, the sampling frequency in our application is approximately 25 Hz. Unfortunately, the application does not guarantee an exact sampling rate. The real sampling frequency is scattering around the ideal with variable deviation. It comes from the mechanism of the operation system. However, it does not cause problem when the device has enough computational capacity and it is not overloaded. In this case the deviation will be negligible. The official Android developer website [23] gives more information about built-in motion sensor handling.

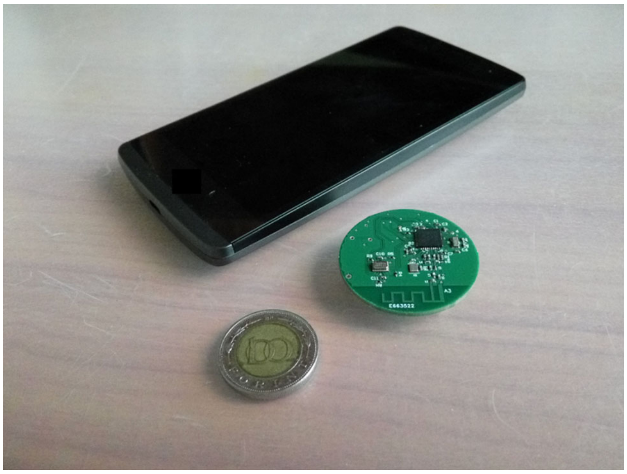


Fig. 2 Size difference between smartphone and a special purpose data collector device (made at University of Debrecen)

2.3 Windowing

Generally, the signal comes from the sensors as a discretized and continuous data flow. To facilitate the activity classification, the continuous signals will be divided into small pieces. Those pieces are called *windows*. The main challenge of this segmentation is to find the correct window size which is suitable for recognition. Researchers followed different approaches in this question. On one hand, short time windows may not provide enough information about the activity. On the other hand, long windows may cover more than one activity in one window. In spite of this ambiguity, most scientists use static window size but some dynamic approaches also exist in the literature [24]. The survey of Lara and Labrador [25] gives a brief description about the advantages and disadvantage of different window sizes.

Generally, the windows size depends on the sampling frequency and it covers one or two seconds wide time period. Gao et al. [15] and Karantonis et al. [22] used 1-s wide windows without overlapping between them. Preece et al. [13] worked with 2-s wide windows which are overlapping with 1 s. Chernbumroong et al. [6] applied a wider window size (3.88 s) with 50% overlap between windows. In this study the window size was approximately 1.3 s (32 samples) with 50% overlap in the training phase. In the test phase we have not used overlap between windows in order to the test process be faster.

2.4 Feature extraction and selection

Many machine learning applications require feature extraction and feature selection. After data segmentation, windows will be the input of the feature extraction methods. Feature extraction tries to take out the relevant information from the raw signal. Instead of normalized sensor data, extracted features are more advantages because a feature characterizes the whole window and the pattern location inside a window does not affect the feature value [26]. An appropriate feature set can significantly improve the classifier performance and makes the classifier model simpler.

In previous papers, different authors extracted different features especially from the time and frequency domains because a well-established feature group does not exist [25]. A small part of scientists [27] used the wavelet transformation for feature extraction instead of time and frequency domains' features. In the paper of Suto et al. [28] the authors tried to collect all relevant feature extraction techniques from the literature of HAR research. Their work also showed that a general feature set does not exist because feature efficiency depends on the movement style of a person. According to their work, 15 feature extraction methods have been implemented in the Android application. Those features can be seen in Table 1. In this study we did not use wavelet transformation for feature extraction because Preece et al. [13] showed that features from the time and frequency domains are more efficient than the wavelet transformation approach. The features from Table 1 were normalized with (1) where $f(i)$ and $f_{\mathrm{norm}}(i)$ are the ith initial and normalized feature values (from the ith window), while μ_f and σ_f are the means and standard deviations of a feature set which come from the training data. Normalization makes features equally important.

$$f_{\mathrm{norm}}(i) = \left(f(i) - \mu_f\right)/\sigma_f \tag{1}$$

In some cases, the number of features is rather huge and some features can be useless. Feature selection is the process of choosing a subset from the original features set according to the distribution of feature vectors or relations (e.g., correlation) between them. It is a frequently used dimensionality reduction technique. Essentially, the goal of all feature selection algorithms independently of their types is to find an appropriate hyperplane in the *feature space* where the class distributions are distinct. The work of Saeys et al. [29] gives additional information about feature selection and its applications in bioinformatics.

One part of HAR researchers did not use feature selection. They selected features from one or two domains without any relation test between them. The other group of researches utilized feature selection. They focused on the supervised category and inside it for the filter and wrapper techniques. Maurer et al. [4] reduced the number of features with the *Correlation based Feature Selection* algorithm. Gou et al. [30] used the information gain (IG). Jatoba et al. [31] chose the *Minimum Redundancy Maximum Relevance* technique. Suto et al. [28] conducted an efficiency investigation between feature selection methods in HAR. They tested a *naïve Bayesian* wrapper method and eight filter-based selection strategies. The selected features were different in the case of each person. It clearly indicates that a generally efficient feature combination which is independent of people does not exist. In their work the *Naïve Bayesian* and the *Chi Square* methods were the most effective. The paper of Damasevicius et al. [32] also contains measurements of feature selection efficiency in HAR. Although the usage a feature selection simplifies the model and speeds up the classification process, sometimes it causes a small accuracy loss [26]. Therefore, our Android application does not use feature selection.

Table 1 Feature extraction methods in the application

Domain	Feature
Time	Mean
	Standard deviation
	Mean absolute deviation
	Root mean square
	Interquartile range
	75th percentile
	Kurtosis
	Signal magnitude area
	Max–min difference
Frequency	Spectral energy
	Spectral entropy
	Spectral centroid
	Principal frequency
Other	Correlation between axes
	Tilt angle

Since the application extracts features from the frequency domain, the fast Fourier transformation (FFT) is essential. During test phase the FFT execution is periodically continuous (on each window). In this case the first step in the FFT is to determine the so-called *phase (or twiddle) factors* and use them as constants during the application run time period. It is possible because the window size is a fix value which is a power of two. Therefore, in the application an improved radix-2 FFT algorithm has been implemented which utilizes all three relations between the phase factors. Equations (2)–(6) describe those relations where W_N^k is the kth phase factor and N is the window size, while *Im* and *Re* refer to the imaginary and real components. By these equations it is enough to calculate and store the first (N/8 + 1) phase factors because their real and imaginary components with the appropriate sign can be substituted into the FFT computation stages. More information about this modified radix-2 FFT and the relations between its twiddle factors can be found in the article of Suto and Oniga [39].

$$W_N^K = -W_N^{K+N/2} \tag{2}$$

$$W_{N_{\mathrm{Im}}}^{K+N/4} = -W_{N_{\mathrm{Re}}}^{k} \tag{3}$$

$$W_{N_{\mathrm{Re}}}^{k+N/4} = W_{N_{\mathrm{Im}}}^{k} \tag{4}$$

$$W_{N_{\mathrm{Im}}}^{k} = -W_{N_{\mathrm{Re}}}^{N/4-k} \tag{5}$$

$$W_{N_{\mathrm{Re}}}^{k} = -W_{N_{\mathrm{Im}}}^{N/4-k} \tag{6}$$

2.5 Machine learning

A proper classification system has to be capable of learn and tolerate noise. In the recent past, researchers have investigated the HAR problem with a wide variety of ML techniques. Several scientists applied parametric methods such as artificial neural network (ANN) convolutional neural network (CNN) and naïve Bayesian, while others tried to use nonparametric algorithms: decision tree (DT), support vector machine (SVM) and k-nearest neighbours (kNN) [25]. Usually the performance of the algorithms has been tested on public databases or on self-recorded data sets. For instance, Yang et al. [19], Oniga and Suto [34] and Su et al. [33] worked on a public database which has been composed at University of California, Berkeley, while in the work of Godfrey et al. [2], Preece et al. [13] and Gao et al. [15] the data acquisition took place in a special environment. In this paper we tried to collect several good reference works from the literature where machine learning algorithm(s) were the classifier. Table 2 contains a list of the collected papers.

As Table 2 demonstrates, different authors used different shallow and deep ML algorithms for HAR. However, some studies have shown that ANN and kNN are the most efficient shallow methods for this purpose. A good comparison between shallow techniques can be found in the papers of Gao et al. [15] and Rahman et al. [35]. In the last decade, the appearance of CNNs caused a breakthrough in several machine leaning topics. The idea of using convolutional and pooling layers has become attractive for HAR community because a CNN can automatically build high-level representations from the raw sensor signal; thus, it eliminates the static feature extraction step from shallow methods [16, 17]. The above reasons motivated us to implement the kNN, ANN and CNN algorithms for this study.

As kNN classifier, the *1NN* scheme has been implemented with the well-known *Euclidean* distance metrics between feature vectors. Actually in all three references in Table 2, the scheme of the kNN classifier was the 1NN. The design of the ANN architecture is based on the article of Suto and Oniga [26] where the authors measured the performance of three different ANN compositions with different hyper-parameters on two public databases. According to their results, the ANN classifier in our implementation has the following settings:

- Two layers (one hidden and one output)
- 40 neurons on the hidden layer
- Gradient descent learning algorithm with momentum
- Stop condition: no improvement in 10 epochs
- Mean square error function with *L2* regularization (9) where M is the number of output neurons, N is the number of samples, y_j is the target, a_j is the activation of the output neuron, while λ and ω refer to the regularization strength and weights, respectively
- Tangent and linear activation functions on the hidden and output layers (10), (11)
- Batch size: 10
- Momentum: 0.15
- Epoch limit: 1000
- Biases were initialized with 0
- Initial weights come from a normal distribution (7) where κ is the inputs of a neuron on the lth layer
- Exponential learning decay as in (8) where α_0 is the initial learning rate, φ is the decay factor, and ε is the epoch counter.

Table 2 Previous recognition rates with machine learning methods in HAR

Classifier	References	Data source	Number of subjects	Recognition rate (%)
kNN	[14]	Not public	5	98.0
	[13]	Not public	20	97.0
	[35]	Not public	8	99.7
DT	[40]	Not public	10	97.3
	[4]	Not public	6	92.8
SVM	[33]	Public database[a]	20	98.5
	[41]	Public database[b]	30	96.0
	[6]	Not public	12	90.2
ANN	[11]	Not public	6	97.9
	[34]	Public database[a]	20	98.1
	[10]	Not public	7	95.0
	[15]	Not public	8	96.8
CNN	[16]	Public database[a]	20	95.9
	[17]	Public database[b]	30	95.2
	[36]	Public database[b]	30	95.8

[a] https://people.eecs.berkeley.edu/~yang/software/WAR

[b] http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

$$\mathbf{W}^{l} \in \mathrm{N}\left(0, \frac{1}{\sqrt{\kappa}}\right) \tag{7}$$

$$\alpha = \alpha_0 \mathrm{e}^{-\varphi\varepsilon} \tag{8}$$

$$\mathrm{E}_1 = \sum_{j}^{\mathrm{M}} \left(y_j - a_j^L\right)^2 + \frac{\lambda}{2\mathrm{N}} \sum_{\omega} \omega^2 \tag{9}$$

$$\sigma_1(\eta) = \frac{\mathrm{e}^{\eta} - \mathrm{e}^{-\eta}}{\mathrm{e}^{\eta} + \mathrm{e}^{-\eta}} \tag{10}$$

$$\sigma_2(\eta) = \eta \tag{11}$$

The CNN construction is also strongly based on earlier articles. For instance, Sheng et al. [16] used two convolutional and pooling layers with 128 and 256 depths and two fully connected layers with 512 and 13 neurons. Jiang et al. [17] tried different constructions and their best architecture has similarly two convolutional and pooling layers with 5 and 10 feature maps and two fully connected layers. The most detailed description about hyper-parameter settings of CNNs can be found in the article of Ronao and Cho [36]. Unlike the previous articles, in this study the CNN input was one-dimensional sensor signal; thus, convolutional layers performed 1 dimensional convolution. The authors have found that after three convolutional layers the performance is decreasing. In addition, after 130 feature maps the performance does not increase.

The above works illustrates that a well-established CNN architecture does not exist in HAR because different authors are trying different approaches. Therefore, we implemented and tested two CNNs (*CNN1* and *CNN2*) with different layer depths and number of neurons on the first fully connected layer. Figures 3 and 4 illustrate the structures of CNN1 and CNN2, respectively. The sensor signal was arranged in two-dimensional form (*number of sensors' axis* × *window size*) because a CNN takes into consideration the spatial relationships between input data. In both CNNs the filter sizes on the convolutional and pooling layers were 2 × 2 with one-sample-long stride on the convolutional layers ($C_{1,2}$) and two-sample-wide strides on the pooling layers ($P_{1,2}$). On a volume in a CNN each neuron has the same filter and bias. More formally, for the i, jth hidden neuron on the lth layer's vth volume, the output is (12) where $N \times M$ is the filter size, V indicates the number of volumes on the previous layer, b is the bias, and ω refers to the weights. The activation function on the convolutional layers is *rectified linear (ReLu)* (13). On the pooling layers the *max-pooling* has been applied which is the most popular in the literature. On the final layer the activation function is soft-max (14) with cross-entropy loss (15) such as in [37, 38].

$$a_{i,j}^{l,v} = \sum_{pv}^{\mathrm{V}} \sigma\left(\sum_{n}^{\mathrm{N}} \sum_{m}^{\mathrm{M}} \omega_{n,m}^{l,v} a_{i+n,j+m}^{l-1,pv} + b^{l,v}\right) \tag{12}$$

$$\sigma_3(\eta) = \max(0, \eta) \tag{13}$$

$$\sigma_4(\eta_{\mathrm{i}}) = \frac{\mathrm{e}^{\eta_{\mathrm{i}}}}{\sum_{\mathrm{j}}^{\mathrm{M}} \mathrm{e}^{\eta_{\mathrm{j}}}} \tag{14}$$

$$\mathrm{E}_2 = -\sum_{j}^{\mathrm{M}} \left[y_j \ln a_j^L + (1 - y_j)\ln\left(1 - a_j^L\right)\right] l + \frac{\lambda}{2\mathrm{N}} \sum_{\omega} \omega^2 \tag{15}$$

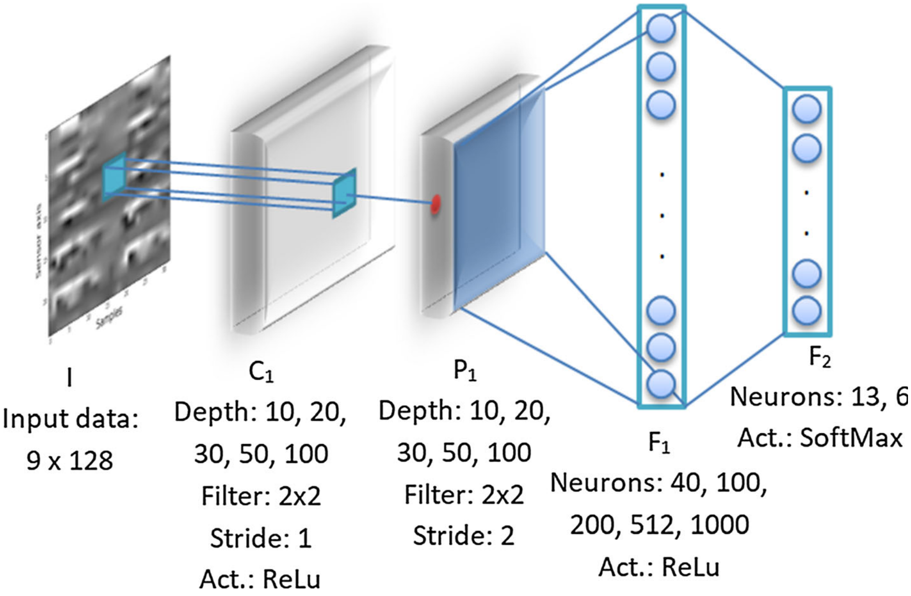


Fig. 3 The first CNN architecture with one convolutional, one pooling, and two fully connected layers. The figure shows the different depth sizes and neurons on the first fully connected layer that have been used in the study

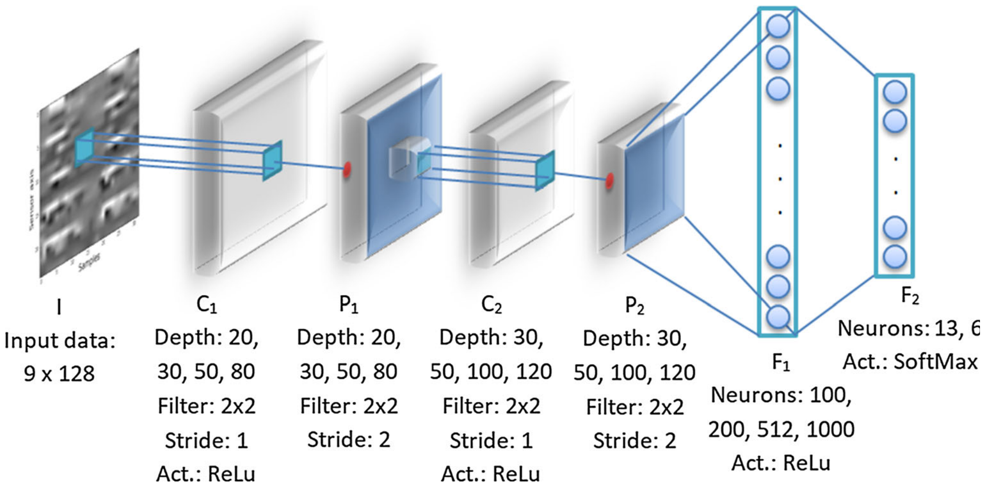


Fig. 4 The second CNN architecture with two convolutional, pooling, and fully connected layers. The depth sizes and number of neurons on the first fully connected layer were also illustrated as in Fig. 3

3 Operation of the application

The application is an improved version of the initial software [43]. First time users should create an account and login to our server. On the registration form they should type in their user name, age and gender. Those parameters will be stored on the server in an SQL database. After registration, the user must login in order to use it. Screenshots of the registration and login forms can be seen in Fig. 5. Thereafter, the pop-up menu on the left side contains the available "*fragments*". On the *data acquisition fragment* the user can collect training data from each activity. Currently, the application is focusing on seven main daily activities, namely

- Cycling
- Running
- Jogging
- Walking
- Sitting
- Standing
- Lying.

The users should designate the time interval while they perform the selected activity. During this time, the software acquires samples from the phone built-in accelerometer and gyroscope and stores the samples into files. When each activity has been performed, the data can be uploaded to our remote server. It makes possible the dynamic data acquisition because data can be acquired independently of us. The *machine learning fragment* performs the feature extraction and classifier training with a selected ML algorithm. Each feature from Table 1 will be extracted from the raw data and normalized by Eq. (1). This normalized feature vector feeds the classifier. On the *"recognition"* frame, the user should perform each activity again until 45 s. During this time interval, the software collects real-time test samples from the accelerometer and gyroscope sensors. The features from the test samples are also normalized by Eq. (1) where the mean and standard deviation come from the training dataset. Finally, the selected ML algorithm predicts the activity class from the normalized feature vector. After a test process, the proportion of correct and incorrect decisions can be seen on a pie chart. Screenshots of the fragments can be seen in Fig. 6. The current version of the application is freely available after an official request. (The download link is password-protected.) Any additional materials can be found on the project's webpage (http://irh.inf.unideb.hu/user/sutoj/har.php).

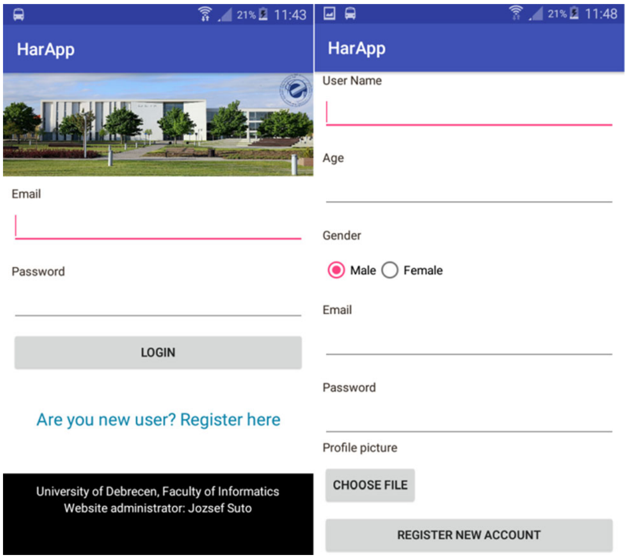


Fig. 5 Screenshots about login and registration forms

4 Results and discussion

4.1 Offline tests

In order to verify the reliability of the ML methods, they have been tested on a public database. The goal of this

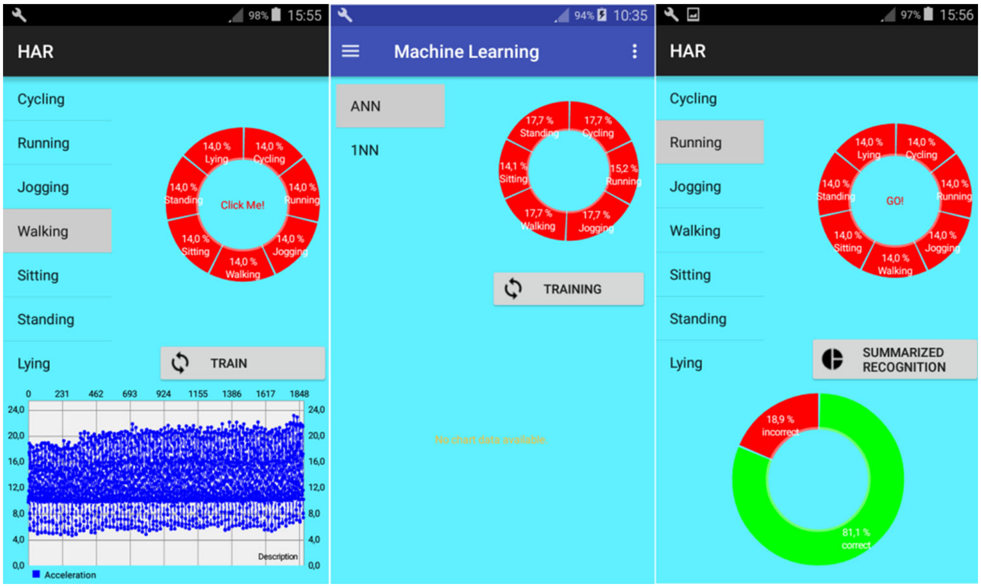


Fig. 6 Screenshots about the main fragments of the application

offline investigation is to verify the correct operation of the classifiers on an independent database. Since the application was written in Java, the program code is easily portable between computer and phone.

The public database [41] was also collected with a single smartphone which contains embedded three-axis accelerometer and gyroscope. Actually it is the ([b]) public database from Table 2 which can be downloaded from the well-known UCI machine learning repository. It his repository, currently it is the seventh most popular dataset. It has been created with a group of 30 participants. They performed the following six daily activities:

- Walking
- Walking upstairs
- Walking downstairs
- Sitting
- Standing
- Laying.

At first the above-mentioned 1NN was tested on the public data and it produced 91.6% recognition rate. The 1NN is the simplest algorithm from the three ML techniques (1NN, ANN and CNN) which does not require any additional hyper-parameters. However, in ANN and CNN several parameters exist which influence their performance. In ANNs the four most significant parameters during neural network training are the number of neurons on the hidden layer(s), regularization strength (λ), learning decay factor (φ) and the initial learning rate (α_0). After some experiments we found that 40 neurons on the hidden layer can be a good choice regarding to the performance and time requirement of the network. In addition, we should highlight that more than one hidden layer is unnecessary. Although in some cases the involvement of additional layers can slightly improve the performance of the network, after two hidden layers the vanishing gradient problem will occur which slows down the learning process. To find an appropriate combination of the remaining three parameters, we performed 100 random hyper-parameter search trials on the database. The search took place on exponential scale where the exponents were randomly drawn from a uniform distribution according to (16). This process has been performed on a laptop with *8 GB* memory and *i5-2.3* GHz processor. The outcome of the parameter search and its time requirement can be seen in Figs. 7 and 8. After hyper-parameter search the following combination produced the highest accuracy: $\alpha_0 = 0.000465$, $\varphi = 0.0000618$, $\lambda = 0.0371$.

$$\alpha_0, \varphi, \lambda \in 10^{U(-4,-1)} \tag{16}$$

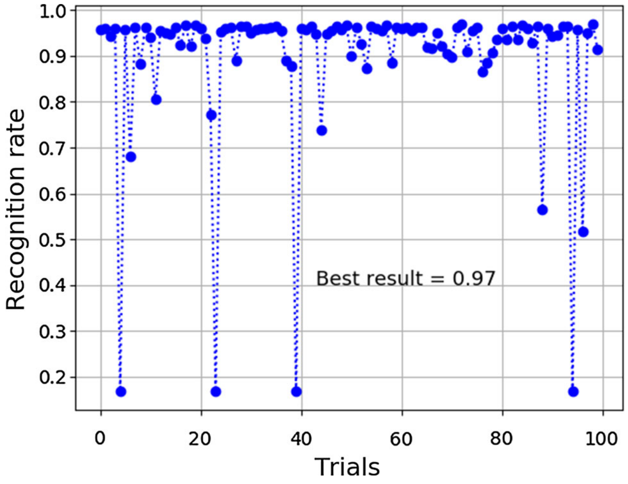


Fig. 7 Result of the random hyper-parameter search after 100 trials

Beyond the network structure, a CNN also requires the same hyper-parameters than an ANN and some other settings such as convolutional and pooling filter sizes and their stride length. To find the optimal parameters to a CNN is still an open question, and it is a slow process with random parameter search because in this case the training time is much longer. Therefore, in the tests, the filter size was 2 × 2 on the convolutional and pooling layers permanently in both CNNs. The stride length is one sample wide on the convolutional layers and two samples wide on the pooling layers. As learning rate and decay factor the same have been used as in the case of ANN. However, the regularization strength was higher than in the previous case ($\lambda = 0.9$) because the overfitting in CNNs is a more serious problem [38]. Since CNNs do not require feature extraction, their input was raw data which come from the sensors in two-dimensional normalized form with (1). In this data matrix a row is a complete window (with 128 samples) from a sensor's axis so the number of rows is equal to the number of axes. The raw data in the database consist of gyroscope, total acceleration and estimated body acceleration values from three axes; therefore, the number of rows in the input data matrix is 9. Tables 3 and 4 contain the best recognition rates and the average time requirement (in second) of both CNNs after three trials. The architecture column in the tables shows the depth of the convolutional layers and the neurons on the first fully connected layer.

The above results demonstrate that our 1NN, ANN and CNN algorithms can produce similarly or better performance than other software which has been used in previous works from Table 2. If we observe the result of the parameter search, the importance of the hyper-parameters in an ANN is clearly visible. Different parameter ensembles produced different accuracies. The recognition rate difference between the best and worst parameter combinations was over 80% (97–16.8%). Moreover, after parameter search the highest recognition rate was 97% which is better than all previous results on the public database.

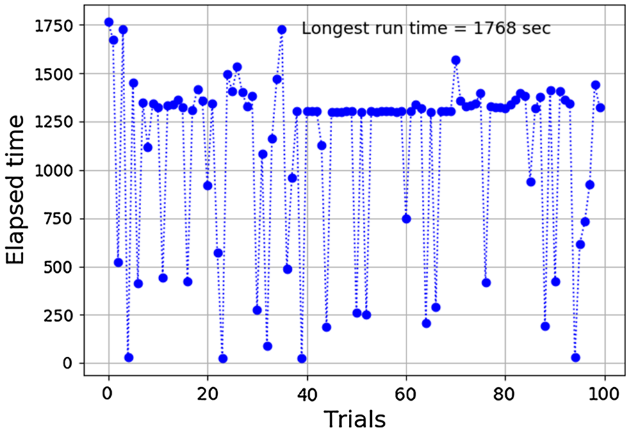


Fig. 8 The training time of the neural network during hyper-parameter search

Table 3 Recognition rates with *CNN1*

Architecture	Rec. rate (%)	Time (s)
10–40	84.6	1615
20–40	85.3	4071
30–100	86.4	6972
50–200	88.3	19,985
50–512	89.8	70,671
100–1000	89.9	286,509

Table 4 Recognition rates with *CNN2*

Architecture	Rec. rate (%)	Time (s)
20-30-100	88.8	16,142
20-50-100	89.5	19,394
30-50-200	90.6	36,747
30-50-512	92.4	62,380
50-100-512	92.8	90,921
80-120-1000	94.2	350,677

In the case of CNNs, more complex structures caused growing recognition rates. However, the best result (94.2%) which has been reached with the deepest CNN2 (80–120–1000) is significantly smaller than the 97%. Probably, the efficiency of CNNs also can be improved by random hyper-parameter search but the training time of a complex CNN is enormous in comparison with a shallow ANN. The training time of the shallow network with the best parameter combination was 1053 s, while the training time of the deepest CNN2 was 350,677 s. On a smartphone the time requirement of a CNN would be much higher thus the usage of CNNs in real-time HAR applications in not a good choice at present. Actually, the main application area of CNNs is the image processing. Already several authors [37, 38, 42] applied CNNs for object recognition or classification (e.g., ImageNet challenge). In those works the scientists created rather deep CNNs (e.g., AlexNet and GoogleLeNet) with several different layers. If we compare the complexity of AlexNet or GoogleLeNet with the CNNs that were used in HAR, we will see that the HAR problem does not require as deep CNNs as object recognition. Consequently, we can suppose that sensor data do not contain such complex features than colour images.

According to the above reasons CNNs were not used in the online tests.

4.2 Online tests

In this study we have followed all principles outlined in the Helsinki Declaration (as revised in 2000) in all the experiments involving human subjects. The online experiments have been carried out with 3 volunteers. The first and third volunteers are 27- and 28-year-old males, while the second is a 17-year-old female. At the beginning of the tests, all participants gave a clear description from the goal of the survey and the operation of the application. We asked the volunteers to collect as many training data as they can. Although the maximum data acquisition time (4 min) does not cause problem in static activities such as in standing or lying but a 4-min-long running activity can be rather exhausting or even impossible for an elderly people. Moreover, the volunteers indicated that more than 4-min-long training data acquisition would be inconvenient. With the maximum (4-min-long training data acquisition from each activity) training data set and 45-s-long test phase we measured the following data preprocessing, training and test times (in second) on a Google Nexus 4 phone:

- ANN training time: 1471 s
- 1NN training time: 0 s
- ANN decision time: 0.001 s
- 1NN decision time: 0.088 s
- Data preprocessing time in training phase: 8.0 s
- Data preprocessing time in test phase: 0.003 s.

Each participant performed the training data acquisitions and the test process alone without supervisor and independently of each other. The outcome of the online tests can be seen in Tables 5 and 6 where the last row is the average classification accuracy.

The measured accuracies in Tables 5 and 6 are not as accurate as in Table 2 or in the above offline investigation. Now the average recognition rates are smaller than before. After a detailed analysis, we found some reasons for the accuracy loss. Our examinations demonstrate that one of the main reasons for performance loss is the large variance in real-time data. In previous articles, most scientists worked on public databases where the deviation between samples is smaller. As was mentioned before, several researches measured high recognition rates on the two popular data sets from Table 2. In both cases the experiments have been performed in indoor environment and under supervision. Probably, the special environment causes more homogeneous data differently from everyday life situations. For instance, Fig. 9 and 10 illustrate two elements wide feature vectors' distributions of three different walk activity records (walkA, walkB and walkC) of the first volunteers from our online experiment and database ([a]) from Table 2. In the database feature vector distributions are more homogeneous than in the online case. The remote feature vectors (from the mean) perhaps are outside of the decision boundary. Moreover, Tables 5 and 6 illustrate that the sitting activity has been recognised poorly. Its reason is the overlap of feature distributions between activities. Figure 11 illustrates such an overlap (with two features) between sitting and standing. In both cases the leg can be in the same position and it prevents the correct classification. It causes accuracy loss because the classifier will not produce reliable decisions on the overlapping area. We can solve it with at least one additional sensor on the appropriate body segment (e.g., on thigh). Finally, the accuracy difference between ANN and 1NN was not as significant as we expected. Although the ANN has better (or more complex) decision boundary generation capabilities and noise tolerance than 1NN, it is not reached much better result. It also shows a particular noise reduction property of feature extraction. Therefore, noise probably is not the main reason for accuracy loss. We thought that one

Table 5 Online results with 1NN

Activities	Participants		
	1 (%)	2 (%)	3 (%)
Lying	51.4	100	68.6
Standing	85.7	88.6	74.3
Sitting	37.1	5.7	74.3
Walking	91.4	74.3	57.1
Jogging	57.1	37.1	57.1
Running	97.3	65.7	85.7
Cycling	88.6	77.1	71.4
Average	72.7	64.1	69.5

Table 6 Online results with ANN

Activities	Participants		
	1 (%)	2 (%)	3 (%)
Lying	94.6	89.2	86.5
Standing	91.9	89.2	94.6
Sitting	16.2	10.8	83.8
Walking	78.4	91.9	89.2
Jogging	8.1	10.8	83.8
Running	97.3	89.2	91.9
Cycling	13.5	97.3	91.9
Average	57.1	68.3	88.8

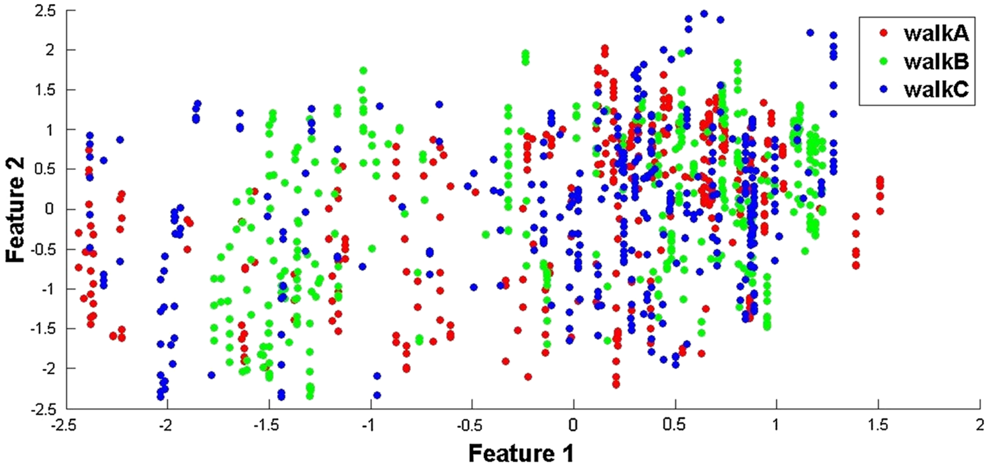


Fig. 9 Walk feature vector distributions from the (a) public database

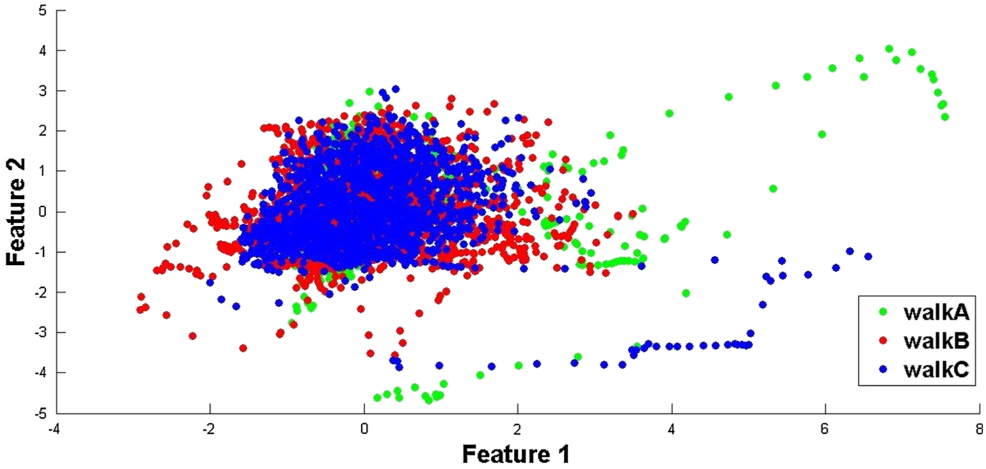
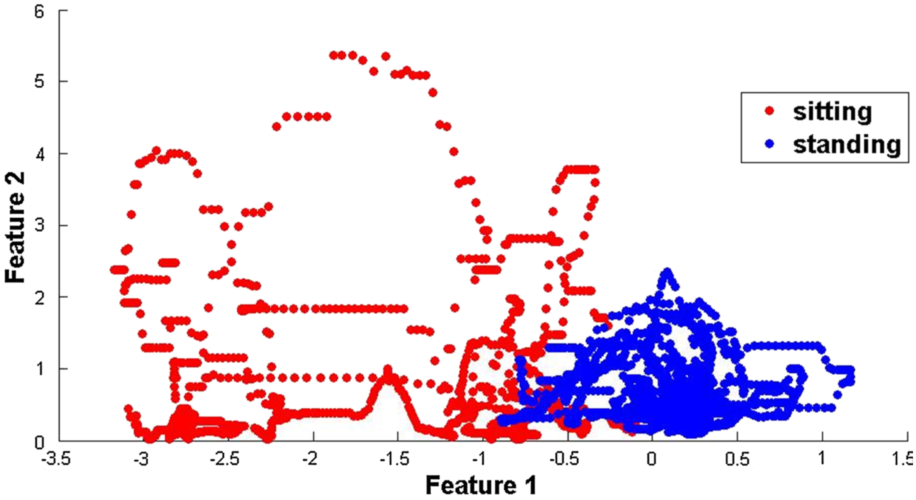


Fig. 10 Walk feature vector distributions from our experiment

Fig. 11 Feature vector overlap between standing and sitting activities



possible reason for the poor ANN performance is the incomplete training data set. Therefore, another experiment also has been performed by participant 1 where the sliding window stride is three samples. It means that the overlap between two adjacent windows is approximately 90%; thus, the training data set is much larger. With the extended training data set ANN produced approximately 20% improvement, while it was less than 10% in the case of 1NN. However, the bigger data set considerably changed the test or training time of the ML algorithms on the same device:

- ANN training time: 8434 s
- 1NN decision time: 0.64 s
- Data preprocessing time in training phase: 16 s.

5 Conclusion

This study examined previous offline activity recognition results in real environment with a self-developed, Android-based application. At the beginning of the article we described the steps of a general activity recognition chain. Moreover, the newest solutions for different activity recognition related questions such as number of sensors, sensor placement, sampling rate, window size, feature extractor and classifier methods also have presented. As reference, we collected some relevant works from the HAR literature in Table 2 independently of the used ML method. According to the outcome of the literature review, 15 feature extractor and 3 ML methods (1NN, ANN, CNN) have been implemented in Java. Before the online investigation, the performance and reliability of our algorithms have been tested on a well-known public database. The offline tests demonstrated that each algorithm can produce similarly or better performance than other software which has been used in previous works. With the best parameter combination the ANN reached 97% recognition rate on the public database which is better than all previous results on it where the researchers did not use parameter search. This result clearly illustrates the importance of hyper-parameters in ANN (and CNN) training because the recognition rate difference between an efficient and an inefficient parameter combination can be significant. Finally, we found that CNNs cannot produce better accuracy than a well-constructed ANN and due to the enormous training time of complex CNNs, currently their usage in real-time HAR applications is not a good choice. Therefore, CNNs have been omitted from the real-time tests.

Although the number of volunteers in this work was relatively small, the online investigation illustrates that there is recognition rate difference between real-time and previous offline HAR results. The first reason behind it is the higher dispersion between feature vectors in real-time data. In previous works usually the experiments have been performed in special environment, under supervision. Therefore, the collected data can be more homogeneous than in the everyday life. However, in online applications significant difference might exist between the training and test data which come from real-life situations. It is unrealistic to collect a complete training data set from all types of activities because a great number of situations exist where test samples will differ from training data. If the training data set is uncomplete the ML algorithm cannot generalize well. We can protect the classifier against this problem with an increase in the training data set. It can be achieved with a longer data acquisition time and with a wider overlapping area between two adjacent windows. We showed that increasing the overlap between windows increased the recognition rate, but it does not solve the uncomplete training data set problem. Unfortunately, training data acquisition is inconvenient, and it is particularly true for the elder population. Therefore, the next problem that HAR community should solve is the training data augmentation. One possible solution for this problem would be a multi-step data collection possibility. In this case a person can acquire data when he wants and the new data will be concatenated to the already existing one.

We can observe the different recognition rates for the same activity of different participants. This can be explained with the individual motion style of people which is changing with the age. Since each human has different motion style, the training data set also has to be individual.

The time requirement of the classifiers with the extended training data set significantly increased, and this causes additional problems. Currently, the 1NN with a great training data set would be unusable on a wearable sensor or sensor network because its test process would be slow, while the ANN's training time would be a very long process. However, our opinion is that the technological development will solve this problem really fast in the near future.

Finally, depending on the activities, one sensor is not always enough because one sensor can generate the same data to different activities. To sum up, based on the current solutions of the HAR literature a relatively acceptable real-time activity recognizer can be constructed but a correct HAR system requires additional improvements.

Acknowledgement This work has been supported by the ÚNKP-17-3-IV New National Excellence Program of the Ministry of Human Capacities.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Paillard-Borg S, Wang HX, Winblad B, Fratiglioni L (2008) Pattern of participation in leisure activities among older people in

relation to their health conditions and contextual factors: a survey in Swedish urban area. Ageing Soc 29:803–821

2. Godfrey A, Conway R, Meagher D, Olaighin G (2008) Direct measurement of human movement by accelerometry. Med Eng Phys 30:1364–1386
3. Sebestyen G, Tirea A, Albert R (2012) Monitoring human activity through portable devices. Carpathian J Electron Comput Eng 5:101–106
4. Maurer U, Smailagic A, Siewiorek DP, Deisher M (2006) Activity recognition and monitoring using multiple sensors on different body positions. In: Proceedings of the international workshop on wearable and implementable body sensor networks. Cambridge, pp 112–116
5. Suto J, Oniga S, Buchman A (2015) Real time human activity monitoring. Annales Mathematicae et Informaticae 44:187–196
6. Chernbumroong S, Cang S, Atkins A, Yu H (2013) Elderly activities recognition and classification for applications in assisted living. Expert Syst Appl 40:1662–1674
7. Shoaib M, Bosch S, Ince OD, Scholten H, Havinga PJM (2015) A survey of online activity recognition using mobile phones. Sensors 15:2059–2085
8. Bayat A, Pomplun M, Tran DA (2014) A study on human activity recognition using accelerometer data from smartphones. Procedia Comput Sci 34:450–457
9. Ayu MA, Ismail SA, Matin AFA, Montoro T (2012) A comparison study of classifier algorithms for mobile-phone's accelerometer based activity recognition. Eng Procedia 41:224–229
10. Yang JY, Wang JS, Chen YP (2008) Using acceleration measurements for activity recognition: an effective learning algorithm for constructing neural classifiers. Pattern Recogn Lett 29:2213–2220
11. Khan AM, Lee YK, Lee SY, Kim TS (2010) A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. IEEE T Inf Technol B 14:1166–1172
12. Oniga S, Suto J (2014) Human activity recognition using neural networks. In: Proceedings of the 15th International Carpathian Control Conference. Velke Karlovice, pp 759–762
13. Preece JS, Goulermas JY, Kenney LPJ, Howard D (2009) A comparison of feature extraction methods for classification of dynamic activities from accelerometer data. IEEE T Bio-Med Eng 56:871–879
14. Duarte F, Lourenco A, Abrantes A (2014) Classification of physical activities using a smart phone: evaluation study using multiple users. Procedia Technol 17:239–247
15. 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
16. Sheng M, Jiang J, Su B, Tang Q, Yahya AA, Wang G (2016) Short-time activity recognition with wearable sensors using convolutional neural networks. In: Proceedings of the 15th ACM SIGGRAPH conference on virtual-reality continuum and its applications in industry. Zhuhai, pp 413–416
17. Jiang W, Yin Z (2015) Human activity recognition using wearable sensors by deep convolutional neural networks. In: Proceedings of the 23th ACM international conference on multimedia, Brisbane, pp 1307–1310
18. Godfrey A, Bourke AK, Olaighin GM, Van Ven de P, Nelson J (2011) Activity classification using a single chest mounted triaxial accelerometer. Med Eng Phys 33:1127–1135
19. Yang AY, Jafari R, Sastry SS, Bajcsy R (2009) Distributed recognition of human actions using wearable motion sensor networks. J Amb Intel Smart En. 1:103–115
20. Ertugrul OF, Kaya Y (2016) Determining the optimal number of body-worn sensors for human activity recognition. Soft Comput 20:1–8
21. Oniga S, Suto J (2016) Activity recognition in adaptive assistive systems using artificial neural networks. Elektron Elektrotech 22:68–72
22. Karantonis DM, Narayanan MR, Mathie M, Lovell NH, Celler BG (2006) Implementation of real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE T Inf Technol B 10:156–167
23. Android developers, API guides, location and sensors, motion sensors. https://developer.android.com/guide/topics/sensors/sensors_motion.html. Accessed 14 Nov 2017
24. Ni Q, Patterson T, Cleland I, Nugent C (2016) Dynamic detection of window starting positions and its implementation within an activity recognition framework. J Biomed Inf 62:171–180
25. Lara OD, Labrador MA (2013) A survey on human activity recognition using wearable sensors. IEEE Commun Surv Tut 15:1192–1209
26. Suto J, Oniga S (2017) Efficiency investigation of artificial neural networks in human activity recognition. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-017-0513-5
27. Ayachi FS, Nguyen HP, Lavigne-Palletier C, Goubault E, Boissy P, Duval C (2016) Wavelet-based algorithm for auto-detection of daily living activities of older adults captured by multiple inertial measurement units (IMUs). Physiol Meas 37:442–461
28. Suto J, Oniga S, Pop-Sitar P (2017) Feature analysis to human activity recognition. Int J Comp Commun 12:116–130
29. Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23:2507–2517
30. Gou Q, Liu B, Chen CW (2016) A two-layer and multi-strategy framework for human activity recognition using smartphone. In: Proceedings of the IEEE international conference on communications, Kuala Lumpur, pp 120–126
31. Jatoba CL, Grobmann U, Kunze U, Ottenbacher J, Stork W (2008) Context-aware mobile health monitoring: evaluation of different pattern recognition methods for classification of physical activity. In: Proceedings of the 30th annual international IEEE EMBS conference, Vancouver, pp 5250–5253
32. Damasevicius R, Vasiljevas M, Salkevicius J, Wozniak M (2016) Human activity recognition in AAL environments using random projections. Comput Math Method M. https://doi.org/10.1155/2016/4073584
33. Su B, Tang Q, Wang G, Sheng M (2016) The recognition of human daily actins with wearable motion sensor systems. In: Transaction on edutainment, 12, Springer, Berlin, pp 68–77
34. Oniga S, Suto J (2015) Optimal recognition method of human activities using artificial neural networks. Meas Sci Rev 15:323–327
35. Rahman HA, Ge D, Faucheur AL, Prioux J, Carrault G (2017) Advanced classification of ambulatory activities using spectral density distances and heart rates. Biomed Signal Poces 34:9–15
36. Ronao CA, Cho SB (2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244
37. Zeiler MD, Fergus M (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision, Zurich, pp 818–833
38. Krizevsky A, Sutskerev I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Neural Information Processing Systems, Nevada, pp 1–9
39. Suto J, Oniga S (2015) A new relation between "twiddle factors" in the fast Fourier transformation. Elektron Elektrotech 21:56–59
40. Kouris I, Koutsouis D (2013) Application of data mining techniques to efficiently monitor chronic diseases using wireless body area networks and smartphones. Univ J Biomed Eng 1:23–31
41. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz L (2013) A public domain dataset for human activity recognition using smartphones. In: Proceedings of 21th European symposium on

artificial neural networks, computational intelligence and machine learning. Bruges, pp 437–442

42. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucker V, Rabinovich R (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, pp 1–9
43. Suto J, Oniga S, Lung C, Orha I (2017) Recognition rate difference between real-time and offline human activity recognition. In: Proceedings of the international conference on internet of things for the global community. Funchal, pp 103–109