Microelectronics Journal xxx (2017) 1–9

Contents lists available at ScienceDirect

Microelectronics Journal

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

Deep learning of smartphone sensor data for personal health assistance

Honggui Li [a,*], Maria Trocan [b]

[a] Yangzhou University, No. 88 South University Road, Yangzhou, 225002, China
[b] Institut Supérieur d'Électronique de Paris, No. 28 Rue Notre-Dame-des-Champs, Paris, 75006, France

ARTICLE INFO

Keywords:
Sensor data
Smartphone
Deep learning
Stacked autoencoders
Personal health

ABSTRACT

This paper strives to construct the relevance between smartphone sensor data and personal health via a deep learning method. Firstly, the data captured by smartphone sensors is categorized into groups by deep stacked autoencoders (SAE), which is a traditional deep learning architecture with multi-layer sparse autoencoders for feature extraction and a softmax layer for classification. For instance, accelerometer data from smartphones can be classified in accordance with different user motion states such as sitting, standing, walking, and running. Secondly, the quantitative correlation between divided sensor data and individual health is built. For example, walking and running for a suitable amount of time are beneficial to human health. Finally, a simulation experiment is devised to verify the performance of the proposed method. A 0.15 percent gain on the overall classification accuracy for automatic feature extraction on the human action recognition (HAR) dataset has been achieved compared with the state of the art methods and 0.5 percent gain for manual feature extraction. It is also demonstrated by the experimental results that smartphone sensor data can reveal the health status of the user to some degree and smartphone can serve as an assistant for human health.

1. Introduction

Recently, smartphone has almost become a necessary commodity for everyone and it can be exploited for calling, chatting, reading, studying, shopping, entertaining, listening to music, watching videos, playing games, making friends, surfing the Internet, and etc. A smartphone is commonly outfitted with diverse sensors, such as accelerometer, gyroscope, thermometer, hygrometer, barometer, cardio-tachometer, dynamometer, magnetometer, sound sensor, image sensor, and so on. It is advantageous for a smartphone user to acquire personal information by various smartphone sensors, which can enhancement user experiences and reflect the intrinsic pertinence between smartphone sensor data and human health to a certain extent. A good case in point is that, accelerometer and gyroscope data of a smartphone reveals user health status to some degree. A smartphone user puts it in his pocket when he goes out for walking or running, and he leaves it on the desk when he stays in his room for sitting or lying. In both situations, smartphone sensor data reflects user motion states. According to accelerometer and gyroscope data, user motion states can be classified into several groups, such as sitting, standing, walking, running, lying, and so forth. It is well known that, running or walking for a reasonable period of time is beneficial to user's health, while sitting for a long period of time in a room can harmful. Thus, individual health status can be partly indicated by smartphone sensor data and smartphone can act as an assistant for human health.

With the swift and violent progress of Internet of things (IoT), big data and cloud computing, wearable device and smartphone hold the formidable power of making use of sensor data for personal health. A good example of this is that, a large number of smartphone applications can conveniently count user motion step, distance, duration, and velocity, on the basis of accelerometer data. A further case is that, a smartphone user can exchange his sensor data with families, friends, colleagues, and classmates in the cloud, such as the IBM cloud for IoT and Emotion Sense [1]. The viewpoint of mobile health (mHealth) has also been presented, and smartphone can exert a major influence on three aspects: ambulatory assessment, behavior monitoring, and understanding and predicting outcomes [1].

Hence, this paper will make great effort to build the correlation between smartphone sensor data and personal health by a deep learning method.

The remaining part of this paper is arranged as follows. The related work is summarized in section 2, the theory framework is established in section 3, the simulation experiment is designed in section 4, and the conclusion is drawn in section 5.

* Corresponding author.
E-mail address: hgli@yzu.edu.cn (H. Li).

https://doi.org/10.1016/j.mejo.2018.01.015
Received 27 March 2017; Received in revised form 6 October 2017; Accepted 16 January 2018
Available online xxxx
0026-2692/

2. Related work

Due to the rapid development of information technology, medical science, biomedical science, and bioinformatics, smartphone sensor data and deep learning methods have already been probed for individual health.

To begin with, sensor data from smartphone, smartwatch, and other wearable devices, has been explored for human health and physiological functions. Human activity recognition (HAR) on the basis of smartphone sensor data is an effective approach to enhance personal health. A. Sinharay presents a smartphone-founded numerical stethoscope for linked health via a straight acoustic coupling method [2]. W.-J. Yi raises an accelerometer and gyroscope founded fall detection and health surveilling system accompanied by a smartphone [3]. G. Horn brings forward a framework for taking advantage of stock devices and smartphones in health systems [4]. A. Sano puts forward a measure of discerning intellectual capability, sleep grade, stress level, and mental health utilizing individuality characteristics, wearable sensors and mobile phones [5]. A. Ojo comes up with a mobile phone texting founded involvement for diabetes and oral health administration [6]. D. D. Mehta advances an approach of mobile voice health surveilling making use of a wearable accelerometer and a smartphone platform [7]. A. A. Farhan proposes a measure of precise anticipation for clinical depression by smartphone sensor data [8]. V. Ahanathapillai employs a smartwatch to extract the parameters of physical activities: activity level, step count, and worn state [9]. E. M. Kidman adopts wearable sensor module to monitor the activities of aesthetic sport athletes [10].

In the next place, deep learning and neural network measures, which maintain the high performance of feature extraction and object classification for sensor data, have also been employed for improving human health or providing the technical potential to strengthen personal heath. A. Sathyanarayana probes the influence of human physical activity on sleep via deep learning founded approaches, such as convolutional neural network (CNN), recurrent neural network (RNN) and long-short term memory (LSTM) [11]. He also suggests that human activity can be adopted for interpreting other health issues, such as obesity, diabetes and cardiac diseases [12]. M. S. Singh proposes a footstep detection method with a pre-trained deep CNNs on 2-dimensional sensor data by transforming the sensor modality to the visual domain [13]. A. Procházka utilizes neural network for the processing of GPS and heart rate signals acquired during long-term physical activities [14]. O. Ťupa exploits neural network to recognize Parkinson's disease via motion tracking and gait feature estimation [15].

In the third place, deep leaning based HAR offers the necessary prerequisite for promoting human health. C. A. Ronao proposes a deep CNN to perform efficient and effective HAR using smartphone sensors by exploiting the inherent characteristics of activities and 1D time-series signals [16]. N. Y. Hammerla explores deep, convolutional, and recurrent approaches for HAR on the data captured with wearable sensors [17]. M. Inoue presents a method of HAR with high throughput from raw accelerometer data applying a deep RNN [18]. F. J. Ordonez raises a generic deep framework for HAR based on convolutional and LSTM recurrent units [19].

In the fourth place, deep learning algorithms have further been realized on smartphones and wearable devices. S. Bhattacharya presents a powerful HAR approach on smartwatches utilizing restricted Boltzmann machines (RBM) [20]. D. Ravi raises a CNN based HAR measure achieved on low-power devices containing smartphones [21]. C. Wang designs a deep learning acceleration unit (DLAU), which is a scalable accelerator architecture for large-scale deep learning networks using field programmable gate array (FPGA) [22].

Last but not the least, new deep learning algorithms continue to spring up. H. Wang tightly integrates deep learning and Bayesian models within a principled probabilistic framework, Bayesian deep learning, to achieve combined intelligence that involves both perception and inference [23]. Q. Zhang brings forward a deep computation model for feature learning on big data, which uses a tensor to model the complex correlations of heterogeneous data [24]. J. Hu puts forward a new deep transfer metric learning (DTML) method to learn a set of hierarchical nonlinear transformations for cross-domain recognition by transferring discriminative knowledge from the labeled source domain to the unlabeled target domain [25]. M. Long comes up with a unified deep adaptation framework for jointly learning transferable representation and classifier to enable scalable domain adaptation, by taking the advantages of both deep learning and optimal two-sample matching [26]. G. Trigeorgis advances a deep semi-non-negative matrix factorization (Semi-NMF) method, which is able to learn hidden representations that allow themselves to be an interpretation of clustering according to the different and unknown attributes of a given dataset [27]. B. Chandra combines adaptive learning rate in conjunction with the concept of Laplacian score for deep learning [28]. He also parameterizes the weight matrix using low rank factorization and periodic functions to speed up deep neural networks (DNN) [29]. Z. Qi focus on the deep learning research based on support vector machine (SVM), proposes an Ex-Adaboost learning strategy, and presents a new deep support vector machine (DeepSVM) [30]. Z. Peng raises a deep boosting framework based on layer-by-layer joint feature boosting and dictionary learning [31]. M. D. Tissera brings forward a method for synthesizing DNN using extreme learning machines (ELMs) as a stack of supervised autoencoders [32].

All in all, this paper attempts to bring forward an evaluation approach of individual health on the basis of smartphone sensor data via conventional deep stacked autoencoders (SAE) [33].

3. Theoretical background

3.1. Global theory framework

The global theory framework of the proposed approach is shown in Fig. 1, and it includes two serial parts: classification and transformation. The classification subassembly divides input data from a smartphone sensor into various motion states, such as running, walking, standing, sitting, sleeping, and so forth, via a deep learning measure. The transformation subassembly tries to build the relevance between smartphone sensor data and personal health. As everyone knows, running and walking for a suitable amount of time facilitate the individual health. The framework can also be depicted as subsequent expressions.

$$\begin{aligned} &\boldsymbol{y} = C(\boldsymbol{X}), \boldsymbol{X} \in R^{N\times P}, \boldsymbol{y} \in R^{1\times P} \\ &\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p & \cdots & \boldsymbol{x}_P \end{bmatrix} \\ &\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_p & \cdots & y_P \end{bmatrix} \\ &z = T(\boldsymbol{y}), z \in R \end{aligned} \tag{1}$$

where:

$\boldsymbol{X}$ is the matrix of input data from a smartphone sensor;
$\boldsymbol{x}_p$ is the p-th column of $\boldsymbol{X}$, which represents an input sample captured by one or multiple smartphone sensors in a time slot, such as 1 or several seconds, and includes many data points, such as 100 or more points;
N is the dimension of an input sample;
P is the total number of input samples by now;
$\boldsymbol{y}$ is the vector of classification result, which indicates the motion states for sensor data, such as accelerometer data or gyroscope data;
y_p is the p-th unit of $\boldsymbol{y}$, which denotes a motion state, such as walking, walking upstairs, walking downstairs, standing, sitting, lying, and so on;

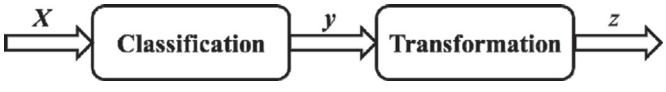

Fig. 1. The theory block diagram of the proposed method.

z is the health index which is corresponding to human health;
C is the classification subassembly which converts $\boldsymbol{X}$ into $\boldsymbol{y}$;
T is the transformation subassembly which converts $\boldsymbol{y}$ into z.

3.2. Motion classification architecture

The theory architecture of the deep learning for human motion classification is illustrated in Fig. 2. It comprises four components: an input layer, multi-layer autoencoders, a classification layer of softmax, and an output layer. The multi-layer autoencoders are exploited for guideless feature extraction, and the softmax layer is adopted for guided target classification. The architecture can be further expressed as following equations.

$$\begin{aligned}
&\boldsymbol{a}_1 = s\left(\boldsymbol{W}_1\boldsymbol{x}_p + \boldsymbol{b}_1\right), \boldsymbol{x}_p \in R^{N\times 1}, \boldsymbol{a}_1 \in R^{M_1\times 1}, \boldsymbol{W}_1 \in R^{M_1\times N}, \boldsymbol{b}_1 \in R^{M_1\times 1}\\
&\boldsymbol{a}_i = s(\boldsymbol{W}_i\boldsymbol{a}_{i-1} + \boldsymbol{b}_i), \boldsymbol{a}_i \in R^{M_i\times 1}, \boldsymbol{W}_i \in R^{M_i\times M_{i-1}}, \boldsymbol{b}_i \in R^{M_i\times 1}, i = 2,3,\cdots,L\\
&y_p = s(\boldsymbol{W}_{L+1}\boldsymbol{a}_L + \boldsymbol{b}_{L+1}), y_p \in R, \boldsymbol{W}_{L+1} \in R^{M_{L+1}\times M_L}, \boldsymbol{b}_{L+1} \in R^{M_{L+1}\times 1}\\
&E_m = \frac{1}{P_0}\sum_{p=1}^{P_0}\left\|y_p - y_{p0}\right\|_2^2, P_0 \in R, y_{p0} \in R\\
&\{\boldsymbol{W}_i, \boldsymbol{b}_i | i = 1,2,\cdots,L+1\} = \underset{\{\boldsymbol{W}_i,\boldsymbol{b}_i|i=1,2,\cdots,L+1\}}{\arg\min} E_m
\end{aligned} \tag{2}$$

where:
$\boldsymbol{x}_p$ is the input of the multi-layer autoencoder, which denotes an input sample;
$\boldsymbol{W}_i$ is the relevant weight matrix of the i-th layer;
$\boldsymbol{b}_i$ is the relevant bias vector of the i-th layer;
$\boldsymbol{a}_i$ is the output of the i-th autoencoder;
y_p is the output of the softmax layer, and is a scalar into which a vector is transformed for expedience;
s is the nonlinear activation function, such as the logistic sigmoid function for the autoencoders and its multiclass generalization, softmax function, for the softmax layer;
N is the dimension of an input sample;
M_i is the total number of neurons in the i-th layer;
L is the total number of the autoencoders;
y_{p0} is the true class of y_p, which represents the motion state for accelerometer data or gyroscope data;
E_m is the cost function of the motion classification infrastructure;
P_0 is the total number of the training samples.

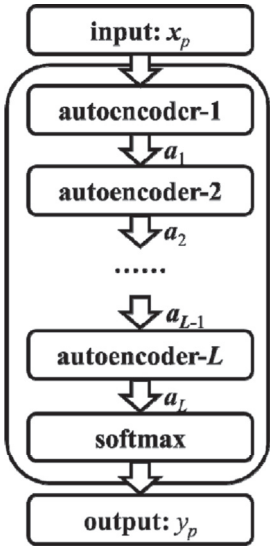

Fig. 2. The theory block diagram of the SAE.

The block diagram of the i-th autoencoder is displayed in Fig. 3, and it consists of three layers: an input layer, a hidden layer and an output layer. It contains all the three layers in the training phase, but it only contains the upper two layers in the testing phase. The upper two layers are known as the i-th encoder, and the lower two layers are known as the i-th decoder. It founds the optimal hidden layer and the related weights by preserving that the output equals the input. The autoencoder can be further described as subsequent formulas.

$$\begin{aligned}
&\boldsymbol{a}_i = s(\boldsymbol{W}_i\boldsymbol{x}_i + \boldsymbol{b}_i)\\
&\boldsymbol{x}_{ri} = s(\boldsymbol{W}_{ri}\boldsymbol{a}_i + \boldsymbol{b}_{ri})\\
&\boldsymbol{W}_{ri} \in R^{M_{i-1}\times M_i}, \boldsymbol{b}_{ri} \in R^{M_{i-1}\times 1}\\
&s(e) = \begin{cases} \mathrm{LogSig}(e) = 1/(1+\exp(-e)) \\ \mathrm{ReLU}(e) = \max\{0, e\} \\ \mathrm{SatLin}(e) = \min(1, \max(0, e)) \end{cases}, e \in R\\
&\boldsymbol{x}_i = \begin{cases} \boldsymbol{x}_p, i = 1 \\ \boldsymbol{a}_{i-1}, i = 2,3,\cdots,L \end{cases}\\
&E_{ri} = \frac{1}{P_0}\sum_{p=1}^{P_0}\|\boldsymbol{x}_{ri} - \boldsymbol{x}_i\|_2^2\\
&\{\boldsymbol{W}_i, \boldsymbol{b}_i, \boldsymbol{W}_{ri}, \boldsymbol{b}_{ri}\} = \underset{\{\boldsymbol{W}_i,\boldsymbol{b}_i,\boldsymbol{W}_{ri},\boldsymbol{b}_{ri}\}}{\arg\min} E_{ri}
\end{aligned} \tag{3}$$

where:

$\boldsymbol{x}_i$ is the input of the i-th encoder, which equals $\boldsymbol{x}_p$ for the first autoencoder and equals $\boldsymbol{a}_{i\text{-}1}$ for other autoencoders;
$\boldsymbol{x}_{ri}$ is the output of i-th decoder, which endeavors to recover $\boldsymbol{x}_i$;
$\boldsymbol{W}_{ri}$ is the related weight matrix of the i-th decoder;
$\boldsymbol{b}_{ri}$ is the related bias vector of the i-th decoder;
s is the nonlinear activation function, which includes logistic sigmoid function (LogSig), rectified linear unit (ReLU), saturating linear transfer function (SatLin), and so on;
e is an element of a vector;
ReLU is the element-wise nonlinear activation function of the i-th autoencoder;
max is the operation which returns the maximum element of a set;
E_{ri} is the reconstruction energy of the i-th autoencoder.

For the situation of sparse autoencoder, its energy functions can be expresses as following equations [34]. Sparsity autoencoder attempts to enforce a constraint on the sparsity of the output from the hidden layer.

$$\begin{aligned}
&E_{li} = \frac{1}{2P_0}\sum_{p=1}^{P_0}\sum_{j=1}^{M_i}\sum_{k=1}^{M_{i-1}}W_{jk}^2\\
&\boldsymbol{\rho}_i = \overline{\boldsymbol{a}_i} = \frac{1}{P_0}\sum_{i=1}^{P_0}\boldsymbol{a}_i = \left[\rho_{i1}\ \ \rho_{i2}\ \ \cdots\ \ \rho_{iM_i}\right]^T\\
&E_{si} = \sum_{p=1}^{P_0}\sum_{j=1}^{M_i}KL\left(\rho_{ij0}\|\rho_{ij}\right) = \frac{1}{P_0}\sum_{p=1}^{P_0}\sum_{j=1}^{M_i}\left(\rho_{ij0}\ln\frac{\rho_{ij0}}{\rho_{ij}} + (1-\rho_{ij0})\ln\frac{1-\rho_{ij0}}{1-\rho_{ij}}\right)\\
&\{\boldsymbol{W}_i, \boldsymbol{b}_i, \boldsymbol{W}_{ri}, \boldsymbol{b}_{ri}\} = \underset{\{\boldsymbol{W}_i,\boldsymbol{b}_i,\boldsymbol{W}_{ri},\boldsymbol{b}_{ri}\}}{\arg\min}\left(E_{ri} + \lambda_{li}E_{li} + \lambda_{si}E_{si}\right)
\end{aligned} \tag{4}$$

where:

E_{li} is the weight-regularization energy of the i-th autoencoder;
$\boldsymbol{\rho}_i$ is the mean of $\boldsymbol{a}_i$;
ρ_{ij} is the j-th component of $\boldsymbol{\rho}_i$;
ρ_{ij0} is the desired value of ρ_{ij};
E_{si} is the sparsity-regularization energy of the i-th autoencoder;
KL is Kullback-Leibler divergence, which is a function for measuring how different two distributions are;
λ_{li} is a coefficient of E_{li};

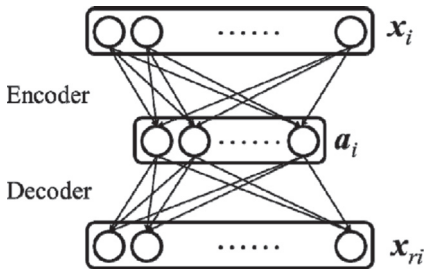

Fig. 3. The block diagram of the *i*-th autoencoder.

λ_{si} is a coefficient of E_{si}.

The block diagram of the softmax classification layer is exhibited in Fig. 4, and it is composed of four layers: an input layer, a hidden layer, a scalar layer, and an output layer. The hidden layer denotes the probabilities of all classes to which the input sample belongs, and the scalar layer converts the input vector into the output scalar for convenience. The softmax classification layer can be further depicted as following expressions.

$$\begin{aligned}
&\boldsymbol{a}_{L+1} = s(\boldsymbol{W}_{L+1}\boldsymbol{a}_L + \boldsymbol{b}_{L+1}) = \mathrm{softmax}(\boldsymbol{W}_{L+1}\boldsymbol{a}_L + \boldsymbol{b}_{L+1}), \boldsymbol{a}_{L+1} \in R^{M_{L+1}\times 1}\\
&\boldsymbol{c}_{L+1} = \boldsymbol{W}_{L+1}\boldsymbol{a}_L + \boldsymbol{b}_{L+1}, \boldsymbol{c}_{L+1} \in R^{M_{L+1}\times 1}\\
&a_{(L+1)i} = \frac{\exp(c_{(L+1)i})}{\sum_{j=1}^{M_{L+1}} \exp(c_{(L+1)j})}\\
&y_p = \mathrm{scalar}(\boldsymbol{a}_{L+1}) = j,\ \text{s.t. } a_{(L+1)j} = \max(\boldsymbol{a}_{L+1})\\
&y_p = s(\boldsymbol{W}_{L+1}\boldsymbol{a}_L + \boldsymbol{b}_{L+1}) = \mathrm{scalar}(\mathrm{softmax}(\boldsymbol{W}_{L+1}\boldsymbol{a}_L + \boldsymbol{b}_{L+1}))\\
&E_s = \frac{1}{P_0}\sum_{p=1}^{P_0} \left\| y_p - y_{p0} \right\|_2^2\\
&\{\boldsymbol{W}_{L+1}, \boldsymbol{b}_{L+1}\} = \underset{\{\boldsymbol{W}_{L+1}, \boldsymbol{b}_{L+1}\}}{\arg\min}\ E_s
\end{aligned} \tag{5}$$

where:

softmax is the nonlinear activation function, which is the multiclass extension of logical sigmoid function and computes the probabilities for all classes to which the input sample pertains;
$\boldsymbol{a}_{L+1}$ is the vector, which denotes the probabilities for all classes to which the input sample belongs;
$a_{(L+1)i}$ is the probability of the *i*-th class to which the input sample belongs;
$\boldsymbol{c}_{L+1}$ is the vector, which represents the sum of the bias vector and the product of the weight matrix and the input vector in the softmax classification layer;

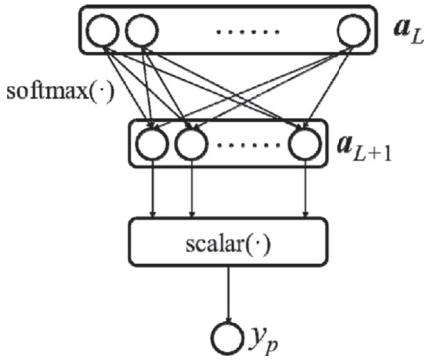

Fig. 4. The block diagram of the softmax classification layer.

scalar is the function, which transforms vector $\boldsymbol{a}_{L+1}$ into a scalar y_p for the convenience of software implementation and returns the index of the element with the maximum probability of $\boldsymbol{a}_{L+1}$.
max is the function, which returns the index of the maximum element of a vector;
E_s is the cost function of softmax classifier.

The classification result y_p can be expressed as subsequent equations, given that it denotes the user motion state. Without doubt, more motion states of smartphone holder can be further listed.

$$y_p = \begin{cases} 1, \text{walking} \\ 2, \text{upstairs} \\ 3, \text{downstairs} \\ 4, \text{sitting} \\ 5, \text{standing} \\ 6, \text{lying} \\ 7, \text{running} \\ 8, \text{jogging} \end{cases} \tag{6}$$

The parameter learning algorithm for the autoencoder layers, the softmax layer, and the whole SAE is the scaled conjugate gradient descent (SCGD), and it can be described as following formulas [35].

$$\begin{aligned}
&\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k + \alpha_k \boldsymbol{p}_k, k = 1, 2, \cdots\\
&s.t.\ \boldsymbol{p}_i \boldsymbol{A} \boldsymbol{p}_j = 0, i \neq j, i, j = 1, \cdots, k\\
&\boldsymbol{\Omega}_{k+1}, \boldsymbol{\Omega}_k \in R^{\Omega\times 1}, \alpha_k \in R, \boldsymbol{p}_k \in R^{\Omega\times 1}, \boldsymbol{A} \in R^{\Omega\times\Omega}
\end{aligned} \tag{7}$$

where:

$\boldsymbol{\Omega}$ is the parameter vector for a given SAE, which includes weights and biases;
α is the step size;
$\boldsymbol{p}$ is the vector of search direction;
$\boldsymbol{A}$ is a non-singular symmetric matrix, and $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\}$ is a conjugate system with respect to $\boldsymbol{A}$;
k is the number of iteration;
Ω is the length of the parameter vector.

3.3. Health transformation architecture

The health transformation part of the proposed method establishes the intrinsic correlation between smartphone sensor data and personal health. The health transform attempts to generate the high health index for the healthy user status and generate the low health index for the unhealthy user status. The health index, which reveals the internal relationship between sensor data and individual health, can be expressed as subsequent equations.

$$z = T(\boldsymbol{y}) = \tanh\left(\frac{q\sum_{p=1}^{P} t(y_p)}{Q}\right) \tag{8}$$

where:

tanh is the hyperbolic tangent function;
q is the adjustment factor of the health transformation;
P is the total number of the samples from a smartphone sensor by now during a day;
t is the health coefficient function, which returns the corresponding health coefficient of y_p;
y_p is also the category number of the p-th sample data from a smartphone sensor during a day;
Q is the expected total number of the healthy samples pertaining to a reference health state, such as walking or running, during a day.

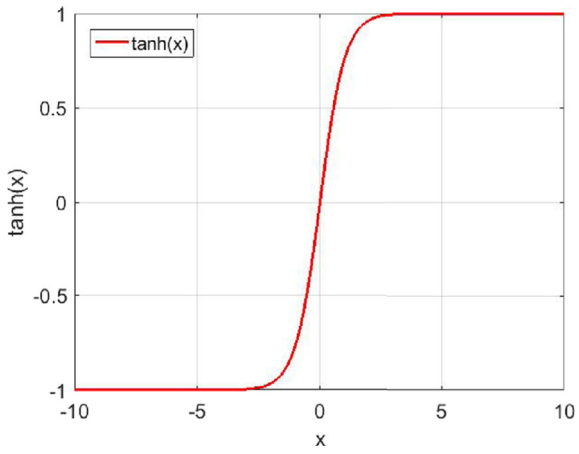

Fig. 5. The plot of hyperbolic tangent function.

The hyperbolic tangent function is illustrated in Fig. 5, and it can be further described as following formulas. The hyperbolic tangent function constraints the value range of the health index, which lies between 0 and 1, because the independent variable of the hyperbolic tangent function for the health index is greater than or equal to 0.

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} = \frac{\sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}}{\sum_{n=0}^{\infty} \frac{x^{2n}}{2n!}} \tag{9}$$

The hyperbolic tangent function has following limits and approximations.

$$\begin{aligned} &\lim_{x\to+\infty} \tanh(x) = \lim_{x\to+\infty} \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} = +1 \\ &\tanh(x) = \frac{\sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}}{\sum_{n=0}^{\infty} \frac{x^{2n}}{2n!}} \to \frac{\frac{x^1}{1!}}{\frac{x^0}{0!}} = x \\ &\lim_{x\to-\infty} \tanh(x) = \lim_{x\to-\infty} \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} = -1 \\ &\tanh(x) \approx \begin{cases} +1, x \geq +2 \\ x, -2 < x < +2 \\ -1, x \leq -2 \end{cases} \end{aligned} \tag{10}$$

The health coefficient function can be depicted as following formulas. Taking accelerometer data and gyroscope data as an example, the faster the motion state is, the higher the health coefficient is. The maximum heath coefficient is 2, and the minimum health coefficient is 0. The health coefficients are reasonable assumptions based on the motion intensity and more quantitative researches are expected.

$$t(y_p) = \begin{cases} 1.0, y_p = 1 \\ 1.1, y_p = 2 \\ 0.9, y_p = 3 \\ 0.0, y_p = 4 \\ 0.0, y_p = 5 \\ 0.0, y_p = 6 \\ 2.0, y_p = 7 \\ 1.5, y_p = 8 \end{cases}, \text{s.t. } y_p = \begin{cases} 1, \text{walking} \\ 2, \text{upstairs} \\ 3, \text{downstairs} \\ 4, \text{sitting} \\ 5, \text{standing} \\ 6, \text{lying} \\ 7, \text{running} \\ 8, \text{jogging} \end{cases} \tag{11}$$

Therefore, three typical health index can be expressed as following equations: high health index, middle health index, and low health index. The healthier the human body is, the higher the health index is. For the value of the health index, 1 denotes high healthy status of a user, 0.5 represents middle healthy status of a user, and 0.1 indicates low healthy status of a user.

$$z = \tanh\left(\frac{q\sum_{p=1}^{P} t(y_p)}{Q}\right) \approx \begin{cases} 1, \frac{q\sum_{p=1}^{P} t(y_p)}{Q} \geq 2 \\ 0.5, \frac{q\sum_{p=1}^{P} t(y_p)}{Q} = 0.5 \\ 0.1, \frac{q\sum_{p=1}^{P} t(y_p)}{Q} = 0.1 \end{cases} \tag{12}$$

4. Experimental results

4.1. Experimental framework

The experimental hardware platform is constructed on a laptop computer with 2.6 GHz dual-core processor and 8.0 GB main memory, and the experimental software platform is a 64-bit operating system of Windows 10 with Matlab 2016a.

There are many open datasets of smartphone sensor, such as the human action recognition (HAR) dataset [36,37], the Actitracker activity prediction dataset of wireless sensor data mining (WISDM) [38,39], and the OPPORTUNITY dataset [40,41]. Only the HAR dataset has been investigated in this paper. The HAR dataset focuses on identifying the human actions implemented by a person given a set of measurements of himself and the neighboring environment. The HAR dataset comprises 30 persons within an age bracket of 19–48 years and 6 activities: walking, walking upstairs, walking downstairs, sitting, standing and lying. The HAR dataset is initially captured by accelerometer and gyroscope on the smartphones worn on the waist of testers. 3-axial linear acceleration and 3-axial angular velocity are captured at a constant rate of 50 Hz. The procedures of data acquisition are video-recorded to label the data manually. The HAR dataset is randomly categorized into two components: 70 percent training data and 30 percent testing data. The total quantity of the training samples is 7352, and the total quantity of the testing samples is 2947. The length of a sample is 768, which includes the data from three axes of accelerometer and three axes of gyroscope. The data size from each axis of accelerometer or gyroscope is 128, which is denoised and resampled in windows of 2.56 s and 50 percent overlap. Each sample of walking motion is coarsely composed of two natural paces. The original data samples of accelerometer and gyroscope are shown in Fig. 6, which consist of 3-axial linear acceleration and 3-axial angular velocity for 2.56 s and 128 sampling points.

Two types of experiments are devised. One is the motion classification

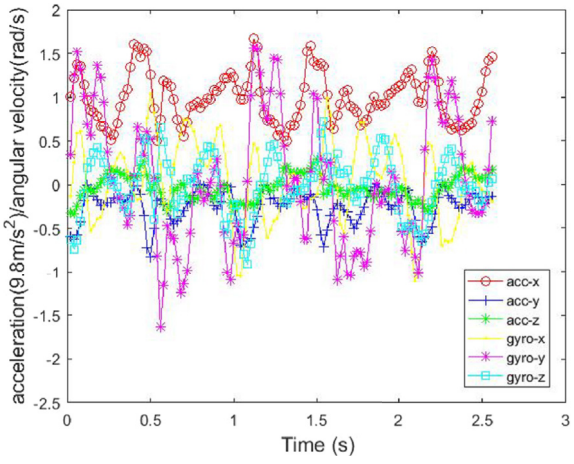

Fig. 6. The original data samples of accelerometer and gyroscope.

experiment, the other is the health transformation experiment. They are detailed below.

4.2. Motion classification experiment

The target of the motion classification experiment is to assess the performance of classifying human motion states founded on the smartphone sensor data via the SAE. The SAE contains an input layer, some latent layers of sparse autoencoders with weight regularization and sparsity regularization, a latent classification layer of softmax and an output layer. The training course of the SAE comprises two stages: the standalone training of each layer and the subtle adjusting of all layers. The training data and testing data of HAR database are utilized for training and testing respectively. The training algorithm for the autoencoder layers, the softmax layer, and the whole DNN is the SCGD. The parameters of the SAE are described in Table 1. For the problem of DNN selection, the SAE is chosen because it holds simple structure, low computation load (without forced GPU), and high performance (with weight and sparsity regularization), compared with other classical DNN architectures. For the problem of DNN initialization, all the weights of the SAE are initialized to values drawn randomly from a Gaussian distribution. For the problem of leaning speed, although it takes several hours for our computer to train the SAE, the testing phase of the SAE is very fast because the learned SAE parameters can be exploited. For the problem of local extrema, the SAE is trained for several times, and the SAE parameters with highest classification accuracy will be retained.

Two sub experiments have been designed. One is the sub experiment with automatic or unsupervised feature extraction, the other is the sub experiment with manual or supervised feature extraction. They are respectively described as follows.

4.2.1. The sub experiment of automatic feature extraction

For the sub experiment of automatic feature extraction, the original sample is utilized as the input vector and the feature vector is gained by the DNN itself. In this situation, deep SAE is employed for unsupervised feature learning and it is a typical practice of deep learning.

In order to improve the classification performance, each component of a training sample or a testing sample is normalized by the related mean and standard deviation of the set of training samples [16]. It can be described as following formulas.

$$\begin{cases} x_{ntr,i} = \dfrac{x_{tr,i} - m_{tr,i}}{\sigma_{tr,i}} \\ x_{nte,i} = \dfrac{x_{te,i} - m_{tr,i}}{\sigma_{tr,i}} \end{cases} \tag{13}$$

where:

$x_{ntr,i}$ is the i-th component of a normalized training sample;
$x_{tr,i}$ is the i-th component of a training sample;
$m_{tr,i}$ is the i-th component of the mean for the set of training samples;
$\sigma_{tr,i}$ is the i-th component of the standard deviation for the set of training samples;
$x_{nte,i}$ is the i-th component of a normalized testing sample;
$x_{te,i}$ is the i-th component of a testing sample.

Table 1
The SAE parameters.

Parameter name	Parameter value
The number of the input layer	1
The number of the autoencoder layer	3
The number of the softmax layer	1
The number of the output layer	1
The coefficient of L2 weight regularization	0.001
The coefficient of sparsity regularization	4
The coefficient of sparsity proportion	0.05
The max epochs	1000
The encoder transfer function	Logistic sigmoid function
The decoder transfer function	Saturating linear transfer function
The training function	SCGD

In order to further upgrade the categorizing performance, a normalized training sample or a normalized testing sample is transformed into its module of Fourier transform [16]. This is because that the module of Fourier transform has the property of translation invariant. It can be depicted as subsequent expressions.

$$\begin{cases} \boldsymbol{x}_{ftr} = |F_6(\boldsymbol{x}_{ntr})| \\ \boldsymbol{x}_{fte} = |F_6(\boldsymbol{x}_{nte})| \end{cases} \tag{14}$$

where:

$\boldsymbol{x}_{ftr}$ is the module of Fourier transform for a normalized training sample;
$\boldsymbol{x}_{ntr}$ is a normalized training sample;
$\boldsymbol{x}_{fte}$ is the module of Fourier transform for a normalized testing sample;
$\boldsymbol{x}_{nte}$ is a normalized testing sample;
F_6 is the six independent Fourier transforms for the normalized data from three axes of accelerometer and three axes of gyroscope;
$|\ |$ is the element-wised module operation for each component of a vector.

For the purpose of exhibiting the experimental results, the confusion matrix can be described as following formulas.

$$\boldsymbol{F} = [f_{ij}], \boldsymbol{F} \in R^{(N_C+1)\times(N_C+1)}, i,j \in \{1,2,\cdots,N_C,N_C+1\}$$

$$f_{ij} = \begin{cases} N_{ij}, \dfrac{N_{ij}}{P}; i,j \in \{1,2,\cdots,N_C\} \\ \dfrac{N_{ii}}{\sum_{j=1}^{N_C} N_{ij}}, 1 - \dfrac{N_{ii}}{\sum_{j=1}^{N_C} N_{ij}}; i \neq N_C+1, j = N_C+1 \\ \dfrac{N_{jj}}{\sum_{i=1}^{N_C} N_{ij}}, 1 - \dfrac{N_{jj}}{\sum_{i=1}^{N_C} N_{ij}}; i = N_C+1, j \neq N_C+1 \\ \dfrac{\sum_{i=1}^{N_C} N_{ii}}{P}, 1 - \dfrac{\sum_{i=1}^{N_C} N_{ii}}{P}; i = N_C+1, j = N_C+1 \end{cases} \tag{15}$$

where:

$\boldsymbol{F}$ is the confusion matrix;
f_{ij} is the unit in the i-th row and the j-th column, which contains two numerical values: the total number and percentage, or the accuracy and inaccuracy;
N_C is the total number of the classes, which is equivalent to the total number of the motion states;
N_{ij} is the total number of the samples, which are predicted as the i-th class and actually belong to the j-th class.

The total quantity of the neurons in the three latent layers of the autoencoders for the HAR dataset is 300, 150 and 75 respectively. The categorizing results of the HAR dataset are exhibited in Table 1, which are a confusion matrix chart for the target class (true class) and output class (predicted classes) respectively. On the confusion matrix chart, the rows denote the predicted class, and the columns represent the true class. The diagonal elements display for how many and what percentage of the samples the trained network correctly forecasts the classes of measurements. That is to say, it shows what percentage of the true and predicted classes match. The off-diagonal elements indicate where the classifier has

Table 2
The confusion matrix of automatic feature extraction on the HAR dataset.

		Target class						
		Walking	Upstairs	Downstairs	Sitting	Standing	Lying	Precision
Output class	Walking	487	2	11	0	0	0	97.4%
		16.5%	0.1%	0.4%	0.0%	0.0%	0.0%	2.6%
	Upstairs	1	458	5	0	0	0	98.7%
		0.0%	15.5%	0.2%	0.0%	0.0%	0.0%	1.3%
	Downstairs	8	11	404	0	0	0	95.5%
		0.3%	0.4%	13.7%	0.0%	0.0%	0.0%	4.5%
	Sitting	0	0	0	432	24	0	94.7%
		0.0%	0.0%	0.0%	14.7%	0.8%	0.0%	5.3%
	Standing	0	0	0	59	508	0	89.6%
		0.0%	0.0%	0.0%	2.0%	17.2%	0.0%	10.4%
	Lying	0	0	0	0	0	537	100%
		0.0%	0.0%	0.0%	0.0%	0.0%	18.2%	0.0%
	Recall	98.2%	97.2%	96.2%	88.0%	95.5%	100%	**95.9%**
		1.8%	2.8%	3.8%	12.0%	4.5%	0.0%	4.1%

made mistakes. The column on the far right of the chart reflects the accuracy and inaccuracy for each predicted class, while the row at the bottom of the chart reflects the accuracy and inaccuracy for each true class. The element in the bottom right of the confusion matrix plot illustrates the whole accuracy and inaccuracy. The whole categorizing accuracy is 95.9 percent, and a 0.15 percent gain on the whole categorizing precision has been attained compared with the benchmark method, CNN [16]. Hence, SAE can obtain higher performance than that of CNN in the area of HAR by suitable normalization and transformation of input data. It is also revealed in Table 2 that, the motion states of walking, upstairs, downstairs and lying can be well recognized with high correct rate, but it is hard to clearly discriminate between sitting state and standing state. This is because that the sensor data of these two states is very similar.

The experimental results of various deep learning methods for unsupervised feature extraction are listed in Table 3. It is demonstrated in Table 3 that, SAE can outperform classical CNN and Deep Belief Network (DBN) in classification accuracy, in the situation of automatic feature extraction and proper data preprocessing.

Table 3
The classification accuracy of deep learning architectures for unsupervised feature extraction.

Architecture	Accuracy
CNN	95.75%
DBN	95.50%
SAE	**95.90%**

4.2.2. The sub experiment of manual feature extraction

For the sub experiment of manual feature extraction, the input vector is the handcrafted feature vector with size 561, which comprises the mean, correlation, signal magnitude area (SMA), auto-regression coefficient, frequency band energy, and frequency skewness of the original sensor data [36,37]. The procedures of data processing and feature extraction are described as follows [36,37]. Firstly, the 3-axial raw signals of accelerometer and gyroscope are filtered utilizing a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Secondly, the acceleration signal is divided into body and gravity acceleration signals utilizing another low pass Butterworth filter with a corner frequency of 0.3 Hz. Thirdly, the body linear acceleration and angular velocity are derived in time to attain Jerk signals. Fourthly, the magnitude of these three-dimensional signals are computed utilizing the Euclidean norm. Finally, a Fast Fourier Transform (FFT) is employed to some of these signals. In such case, SAE is only exploited for classification and it is a mere deep learning based classifier. The total quantity of the neurons in the three hidden layers of the autoencoders for the HAR dataset is 280, 140 and 70 respectively. The classification results of manual feature extraction on the HAR dataset are demonstrated in Table 4. The whole classification precision is 96.5 percent, and a 0.5 percent gain on the whole classification precision has been achieved compared with the baseline method, multiclass support vector machine (MSVM) [36,37]. Therefore, SAE based classifier can achieve higher performance than that of classical support vector machine (SVM). It is also indicated in Tables 2–5 that, the classification accuracy of manual feature extraction is superior to that of automatic feature extraction, but the gap is very small. It is believed that, deep learning based unsupervised feature extraction for HAR will overmatch manual

Table 4
The confusion matrix of manual feature extraction on the HAR dataset.

		Target class						
		Walking	Upstairs	Downstairs	Sitting	Standing	Lying	Precision
Output class	Walking	492	24	3	0	0	0	94.8%
		16.7%	0.8%	0.1%	0.0%	0.0%	0.0%	5.2%
	Upstairs	2	447	17	0	1	0	95.7%
		0.1%	15.2%	0.6%	0.0%	0.0%	0.0%	4.3%
	Downstairs	2	0	400	0	0	0	99.5%
		0.1%	0.0%	13.6%	0.0%	0.0%	0.0%	0.5%
	Sitting	0	0	0	447	10	0	97.8%
		0.0%	0.0%	0.0%	15.2%	0.3%	0.0%	2.2%
	Standing	0	0	0	44	521	0	92.2%
		0.0%	0.0%	0.0%	1.5%	17.7%	0.0%	7.8%
	Lying	0	0	0	0	0	537	100%
		0.0%	0.0%	0.0%	0.0%	0.0%	18.2%	0.0%
	Recall	99.2%	94.9%	95.2%	91.0%	97.9%	100%	**96.5%**
		0.8%	5.1%	4.8%	9.0%	2.1%	0%	3.5%

Table 5
The classification accuracy of deep learning based or conventional classifiers for supervised feature extraction.

Classifier	Accuracy
SVM	96.00%
DBN	95.80%
SAE	**96.50%**

feature extraction in the near future.

The experimental results of different deep learning based or conventional classifiers for supervised feature extraction are exhibited in Table 5. It is manifested in Table 5 that, SAE is superior to traditional SVM and DBN in classification precision, in the case of manual feature extraction.

4.3. Health transformation experiment

The destination of the health transformation experiment is to convert the motion states into user health status, such as health index. Three representative computing instances are devised: the high health index, the middle health index and the low health index. It is sound to hypothesize that Q equals 3000, which is approximately related to 6000 healthy human paces during a day. It is also testified by our experiment that a person averagely needs 6000 slow steps or equivalent amount of exercise to retain his/her health in one day.

The case of the high healthy index is depicted as subsequent expressions, and z is 1, which denotes the high status of the body health.

$$q = 2, \sum_{p=1}^{P} t(y_p) = 3000, Q = 3000$$
$$\Rightarrow z = \tanh\left(\frac{q\sum_{p=1}^{P} t(y_p)}{Q}\right) = \tanh\left(\frac{2\times 3000}{3000}\right) = \tanh(2) \approx 1 \quad (16)$$

The case of the middle healthy index is expressed as following equations, and z is 0.5, which indicates the middle status of the body health.

$$q = 2, \sum_{p=1}^{P} t(y_p) = 750, Q = 3000$$
$$\Rightarrow z = \tanh\left(\frac{q\sum_{p=1}^{P} t(y_p)}{Q}\right) = \tanh\left(\frac{2\times 750}{3000}\right) = \tanh(0.5) \approx 0.5 \quad (17)$$

The case of the low healthy index is described as subsequent formulas, and z is 0.1, which represents the low status of the body health.

$$q = 2, \sum_{p=1}^{P} t(y_p) = 150, Q = 3000$$
$$\Rightarrow z = \tanh\left(\frac{q\sum_{p=1}^{P} t(y_p)}{Q}\right) = \tanh\left(\frac{2\times 150}{3000}\right) = \tanh(0.1) \approx 0.1 \quad (18)$$

5. Conclusion

An evaluation approach of user health by deep learning and smartphone sensor data is presented in this paper. In the first place, an architecture of SAE is established, which is composed of multi-layer autoencoders and a classification layer of softmax. The accelerometer and gyroscope data from smartphones is divided by the SAE into several motion classes: walking, walking upstairs, walking downstairs, sitting, standing and lying. In the second place, the relationship between the motion states and the personal health is constructed, and three typical health indexes can be deduced from the correlation model: the high healthy index, the middle healthy index and the low healthy index. In the third place, the rationality and availability of the proposed method are assessed by the experimental results. A 0.15 percent gain on the global categorizing accuracy for automatic feature extraction on the HAR dataset has been achieved compared with the state of the art algorithms, while a 0.5 percent gain for manual feature extraction. In summary, this paper proposes an efficient measure for reflecting the status of individual health by taking advantage of the data from smartphone sensors, and smartphone can do duty for an assistant for human health.

In our future work, more precise correlation between smartphone sensor data and human health will be investigated, more genres of sensor data from smartphones and wearable devices will also be researched, and more sophisticated infrastructures of deep learning will be further studied.

References

[1] G.M. Sandstrom, N. Lathia, C. Mascolo, P.J. Rentfrow, Opportunities for smartphones in clinical care: the future of mobile mood monitoring, J. Clin. Psychiatr. 77 (2) (2016) 135–137.
[2] A. Sinharay, D. Ghosh, P. Deshpande, Smartphone based digital stethoscope for connected health - a direct acoustic coupling technique, in: Proceedings of the 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016, pp. 193–198.
[3] W.-J. Yi, O. Sarkar, T. Gonnot, E. Monsef, J. Saniie, 6LoWPAN-enabled fall detection and health monitoring system with Android smartphone, in: Proceedings of the 2016 IEEE International Conference on Electro Information Technology (EIT), 2016, pp. 174–178.
[4] G. Horn, F. Eliassen, A. Taherkordi, S. Venticinque, B.D. Martino, M. Bücher, L. Wood, An architecture for using commodity devices and smart phones in health systems, in: Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), 2016, pp. 255–260.
[5] A. Sano, A.J. Phillips, A.Z. Yu, A.W. McHill, S. Taylor, N. Jaques, C.A. Czeisler, E.B. Klerman, R.W. Picard, Recognizing academic performance, sleep quality, stress level, and mental health using personality traits, wearable sensors and mobile phones, in: Proceedings of the 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2015, pp. 1–6.
[6] A. Ojo, S. Chatterjee, H.W. Neighbors, G.A. Piatt, S. Moulik, B.D. Neighbors, J. Abelson, C. Krenz, D. Jones, OH-BUDDY: mobile phone texting based intervention for diabetes and oral health management, in: Proceedings of the 2015 48th Hawaii International Conference on System Sciences, 2015, pp. 803–813.
[7] D.D. Mehta, M. Zañartu, S.W. Feng, H.A. Cheyne II, R.E. Hillman, Mobile voice health monitoring using a wearable accelerometer sensor and a smartphone platform, IEEE (Inst. Electr. Electron. Eng.) Trans. Biomed. Eng. 59 (11) (2012) 3090–3096.
[8] A.A. Farhan, C. Yue, R. Morillo, S. Ware, J. Lu, J. Bi, J. Kamath, A. Russell, A. Bamis, B. Wang, Behavior vs. introspection: refining prediction of clinical depression via smartphone sensing data, in: Proceedings of the 2016 IEEE Wireless Health Conference, 2016, pp. 1–8.
[9] V. Ahanathapillai, J.D. Amor, Z. Goodwin, C.J. James, Preliminary study on activity monitoring using an android smart-watch, Healthcare Technol. Lett. 2 (1) (2015) 34–39.
[10] E.M. Kidman, M.J.A. D'Souza, S.P.N. Singh, A wearable device with inertial motion tracking and vibro-tactile feedback for aesthetic sport athletes Diving Coach Monitor, in: Proceedings of the 2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS), 2016, pp. 1–6.
[11] A. Sathyanarayana, S. Joty, L. Fernandez-Luque, F. Ofli, J. Srivastava, A. Elmagarmid, S. Taheri, T. Arora, Impact of Physical Activity on Sleep: a Deep Learning Based Exploration, arXiv:1607.07034, 2016, pp. 1–10.
[12] A. Sathyanarayana, F. Ofli, L. Fernandes-Luque, J. Srivastava, A. Elmagarmid, T. Arora, S. Taheri, Robust automated human activity recognition and its application to sleep research, in: Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016, pp. 495–502.
[13] M.S. Singh, V. Pondenkandath, B. Zhouy, P. Lukowiczz, M. Liwicki, Transforming sensor data to the image domain for deep learning - an application to footstep detection, in: Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2665–2672.
[14] A. Procházka, S. Vaseghi, H. Charvátová, O. Ťupa, O. Vyšata, Cycling segments multimodal analysis and classification using neural networks, Appl. Sci. 7 (6) (2017) 581–591.
[15] O. Ťupa, A. Procházka, O. Vyšata, M. Schätz, J. Mareš, M. Vališ, V. Mařík, Motion tracking and gait feature estimation for recognising Parkinson's disease using MS kinect, Biomed. Eng. Online 14 (1) (2015) 97–116.

[16] C.A. Ronao, S.-B. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, Expert Syst. Appl. 59 (2016) 235–244.
[17] N.Y. Hammerla, S. Halloran, T. Plotz, Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables, arXiv:1604.08880, 2016, pp. 1–8.
[18] M. Inoue, S. Inoue, T. Nishida, Deep Recurrent Neural Network for Mobile Human Activity Recognition with High Throughput, arXiv:1611.03607, 2016, pp. 1–10.
[19] F.J. Ordonez, D. Roggen, Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (1) (2016) 1–25.
[20] S. Bhattacharya, N.D. Lane, From smart to deep: robust activity recognition on smartwatches using deep learning, in: Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 2016, pp. 1–6.
[21] D. Ravi, C. Wong, B. Lo, G.-Z. Yang, Deep learning for human activity recognition: a resource efficient implementation on low-power devices, in: Proceedings of the 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2016, pp. 71–76.
[22] C. Wang, L. Gong, Q. Yu, X. Li, Y. Xie, X. Zhou, DLAU: a scalable deep learning accelerator unit on FPGA, IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 36 (3) (2017) 513–517.
[23] H. Wang, D.-Y. Yeung, Towards Bayesian deep learning: a framework and some existing methods, IEEE Trans. Knowl. Data Eng. 28 (12) (2016) 3395–3408.
[24] Q. Zhang, L.T. Yang, Z. Chen, Deep computation model for unsupervised feature learning on big data, IEEE Transac. Services Comput. 9 (1) (2016) 161–171.
[25] J. Hu, J. Lu, Y.-P. Tan, J. Zhou, Deep transfer metric learning, IEEE Trans. Image Process. 25 (12) (2016) 5576–5588.
[26] M. Long, J. Wang, Y. Cao, J. Sun, P.S. Yu, Deep learning of transferable representation for scalable domain adaptation, IEEE Trans. Knowl. Data Eng. 28 (8) (2016) 2027–2040.
[27] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B.W. Schuller, A deep matrix factorization method for learning attribute representations, IEEE Trans. Pattern Anal. Mach. Intell. 39 (3) (2017) 417–429.
[28] B. Chandra, R.K. Sharma, Deep learning with adaptive learning rate using Laplacian score, Expert Syst. Appl. 63 (2016) 1–7.
[29] B. Chandra, R.K. Sharma, Fast learning in deep neural networks, Neurocomputing 171 (2016) 1205–1215.
[30] Z. Qi, B. Wang, Y. Tian, P. Zhang, When ensemble learning meets deep learning: a new deep support vector machine for classification, Knowl. Base Syst. 107 (2016) 54–60.
[31] Z. Peng, Y. Li, Z. Cai, L. Lin, Deep boosting: joint feature selection and analysis dictionary learning in hierarchy, Neurocomputing 178 (2016) 36–45.
[32] M.D. Tissera, M.D. McDonnell, Deep extreme learning machines: supervised autoencoding architecture for classification, Neurocomputing 174 (2016) 42–49.
[33] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[34] B.A. Olshausen, D.J. Field, Sparse Coding with an Overcomplete Basis Set: a Strategy Employed by V1, Vision Research 37, 1997, pp. 3311–3325.
[35] M.F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Network. 6 (4) (1993) 525–533.
[36] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, in: Proceedings of the 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2013, pp. 437–442.
[37] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, Energy efficient smartphone-based activity recognition using fixed-point arithmetic, J. Univ. Comput. Sci. Spec. Issue Ambient Assist. Living: Homecare 19 (9) (2013) 1295–1314.
[38] J.W. Lockhart, G.M. Weiss, J.C. Xue, S.T. Gallagher, A.B. Grosner, T.T. Pulickal, Design considerations for the WISDM smart phone-based sensor mining architecture, in: Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data (WKDD), 2011, pp. 25–33.
[39] G.M. Weiss, J.W. Lockhart, The impact of personalization on smartphone-based activity recognition, in: Proceedings of the AAAI-12 Workshop on Activity Context Representation: Techniques and Languages, 2012, pp. 98–104.
[40] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, J. d. R. Millàn, Collecting complex activity data sets in highly rich networked sensor environments, in: Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS), 2010, pp. 233–240.
[41] R. Chavarriaga, H. Saghaa, A. Calatronib, S.T. Digumartia, G. Trösterb, J. d. R. Millána, D. Roggenb, The opportunity challenge: a benchmark database for on-body sensor-based activity recognition, Pattern Recogn. Lett. 34 (15) (2013) 2033–2042.

Honggui Li was born in Yangzhou, China, in 1971. He received a B.S. degree in electronic science and technology from Yangzhou University in 1994 and received a Ph.D. degree in mechatronic engineering from Nanjing University of Science and Technology in 1999. He is a senior member of the Chinese Institute of Electronics. He is a visiting scholar and a post-doctoral fellow in Institut Supérieur d'Électronique de Paris in 2016. He is an associate professor of electronic science and technology and a postgraduate supervisor of electronic science and technology at Yangzhou University. He is a reviewer for some Springer international journals. He is the author of over 30 refereed journal and conference articles. His current research interests include computer vision, machine learning, and embedded computing.

Maria Trocan received her B.Eng. in Electrical Engineering and Computer Science from Politehnica University of Bucharest in 2004, her Ph.D. from Telecom ParisTech in 2007, and her Habilitation to Lead Researches (HDR) from Pierre and Marie Curie University in 2014. Since May 2009 she is Associate Professor with the Signal, Image and Telecommunications Department at Institut Superieur d'Electronique de Paris (ISEP), where since October 2011 she is responsible for the Signal Processing Graduate Program. Her current research interests focus on image and video analysis and compression, sparse representations and wavelet-based processing techniques.