Locomotion Activity Recognition Using Stacked Denoising Autoencoders

Fuqiang Gu, *Student Member, IEEE*, Kourosh Khoshelham, Shahrokh Valaee, *Senior Member, IEEE*, Jianga Shang, *Member, IEEE*, and Rui Zhang

***Abstract*—Locomotion activity recognition (LAR) is important for a number of applications, such as indoor localization, fitness tracking, and aged care. Existing methods usually use handcrafted features, which requires expert knowledge and is laborious, and the achieved result might still be suboptimal. To relieve the burden of designing and selecting features, we propose a deep learning method for LAR by using data from multiple sensors available on most smart devices. Experimental results show that the proposed method, which learns useful features automatically, outperforms conventional classifiers that require the hand-engineering of features. We also show that the combination of sensor data from four sensors (accelerometer, gyroscope, magnetometer, and barometer) achieves a higher accuracy than other combinations or individual sensors.**

***Index Terms*—Activity recognition, autoencoder, deep learning, motion state recognition, neural network, smartphone sensor.**

I. Introduction

HUMAN activity recognition is important for a large number of applications, such as indoor localization, transportation mode detection, smart hospitals or factories, home automation, targeted advertising, and pervasive gaming. Methods for activity recognition can be divided into two categories: 1) ambient sensing methods and 2) mobile/wearable sensing methods. The ambient sensing methods make use of an infrastructure (e.g., a network of video cameras or wireless access points) to sense human activities. For instance, nowadays closed-circuit television systems can be used to detect human activities by exploiting computer vision techniques, such as feature extraction, movement detection, and trajectory analysis [2]. Also, human activities can be recognized by utilizing wireless signal attenuation, fading and propagation characteristics within the coverage area of wireless transceivers [3]. Mobile/wearable sensing methods are usually based on sensors built in modern smart devices [4], such as accelerometer, gyroscope, magnetometer, and barometer. Nowadays, the mobile/wearable sensing methods have become very popular due to two reasons. First, they have no coverage limit and hence can work anywhere. Second, they are more widely available because of the advent of sensor-rich smart devices, such as smartphones, smart watches, and smart glasses.

Human activities can be categorized into different types [5], including locomotion (e.g., walking, running, standing, and still), exercise (e.g., cycling and playing soccer), health related activities (e.g., falls, rehabilitation, and following routines), daily activities (e.g., shopping, using computer, sleeping, going to work, and attending a meeting), and so on. In this paper, we focus on the recognition of locomotion activities, which is important for many applications, such as indoor localization (especially using pedestrian dead reckoning method), energy expenditure, fitness tracking, and aged care.

Generally, locomotion activity recognition (LAR) involves data preprocessing, segmentation, feature extraction, feature selection, modeling, and classification. The collected sensor data are preprocessed to filter out random noise, and then segmented into sequences. From these sequences, different features are extracted and the relevant ones are selected. Then, the selected features are used to train a classification model, after which new unlabeled data can be classified using the trained model. A major challenge of LAR is the hand-engineering and selection of features. The classification performance depends highly on the relevance of the used features. For example, using only the magnitude of acceleration, the classifier might confuse walking with running when the user swings her phone while walking, whereas this can be correctly classified by leveraging frequency features. To ensure a classifier works properly, one can extract as many features as possible from different domains, such as statistical domain, frequency domain, and time domain. However, more features do not necessarily improve the classification accuracy, but will increase computational cost. Therefore, it is important to select appropriate features through certain feature selection methods, such as filters and wrappers [6]. Although the selected

Manuscript received January 24, 2018; revised March 1, 2018; accepted March 20, 2018. Date of publication April 4, 2018; date of current version June 8, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 41271440 and in part by the China Scholarship Council—University of Melbourne Research Scholarship under Grant CSC 201408420117. An earlier version of this paper appeared in the 28th Annual IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (IEEE PIMRC 2017) [1]. *(Corresponding author: Fuqiang Gu.)*

F. Gu and K. Khoshelham are with the Department of Infrastructure Engineering, University of Melbourne, Parkville, VIC 3000, Australia (e-mail: fuqiangg@student.unimelb.edu.au; k.khoshelham@unimelb.edu.au).

S. Valaee is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: valaee@ece.utoronto.ca).

J. Shang is with the Faculty of Information Engineering, China University of Geosciences and the National Engineering Research Center for Geographic Information System, Wuhan 430074, China (e-mail: jgshang@cug.edu.cn).

R. Zhang is with the Department of Computing Information Systems, University of Melbourne, Parkville, VIC 3000, Australia (e-mail: rui.zhang@unimelb.edu.au).

Digital Object Identifier 10.1109/JIOT.2018.2823084

features can improve the classification rate, but existing feature selection methods are not robust enough [7]. A small data perturbation will lead to quite different sets of features. Overall, LAR using handcrafted features requires laborious human intervention and expert knowledge, and might still achieve suboptimal performance.

In this paper, we propose a deep learning method for LAR, which consists of stacked denoising autoencoders [8]. This is motivated by the success of deep learning in different domains [9]–[14], especially in acoustic modeling since acoustic signals are similar to smartphone sensor signals in terms of temporal fluctuations. Stacked autoencoders is a commonly used feature learning method, which is capable of learning useful features in an unsupervised manner [15]. To make the learned features more robust, we use stacked denoising autoencoders in this paper. To the best of our knowledge, this is the first work to apply stacked denoising autoencoders for LAR.

Compared with conventional machine learning techniques using handcrafted features, the main advantage of our method is the automation of feature learning, which eliminates the need for the hand-engineering of features. Moreover, the proposed method can make use of unlabeled data for model fitting in an unsupervised pretraining phase, which is especially useful when labeled data are scarce. In addition, unlike existing works based on smartphone, which use accelerometer data only or few types of sensor data, our method takes advantage of four types of sensor data, i.e., data from accelerometer, gyroscope, magnetometer, and barometer. Experimental results show that the combination of the four types of sensor data achieves better performance than that using few types of sensor data. As will be shown, our method outperforms previous works, achieving a high recognition rate of about 94% as measured by F-measure.

The remainder of this paper is organized as follows. Section II reviews the related literature. In Section III, we present the architecture of the proposed method, and then introduce locomotion activities of interest, data preprocessing, segmentation, and the deep LAR model. The experiments and results are presented and discussed in Section IV. Finally, the conclusions are drawn in Section V.

II. Related Work

A. Locomotion Activity Recognition

LAR can be categorized as ambient sensing methods and mobile/wearable sensing methods. Ambient sensing methods are based on an infrastructure, such as a network of video cameras, wireless access points, and cell towers. Vision-based activity recognition, which has been studied for decades [2], [16], involves labeling image sequences or videos with different activities. Global system for mobile (GSM) has also been used for recognizing human activities by using the signal fluctuation information between the receiver and the cell tower [17], which can distinguish human activities, such as driving, walking, and remaining stationary. Similar to GSM, WiFi signal strength has been applied for recognizing activities, gestures, and environmental situations [18]. Recently, many researchers have demonstrated the feasibility of detecting activities from WiFi channel status information (CSI) [19], [20]. The advantage of these ambient sensing methods is their nonintrusiveness, making these methods especially suitable for security-related applications such as intrusion detection. However, their coverage is quite limited and they are hardly capable of capturing people's continuous activities.

Compared with ambient sensing methods, mobile/wearable sensing methods which are based on mobile/wearable devices, such as smartphones, smart watches, and smart bands, do not have the coverage limitation. Nowadays mobile/wearable sensing methods [21] have become very popular due to the advent of mobile/wearable computing era. Modern smart devices integrate a number of sensors that can be used for activity recognition, including accelerometer, gyroscope, magnetometer, barometer, microphone, and light sensor. Among these sensors built in smart devices, the accelerometer is one of the most widely used [22] since it allows recognizing a variety of activities. Other motion-related sensors [6] such as gyroscope, magnetometer, and barometer, are often jointly used to complement the accelerometer. In this paper, we focus on LAR using smartphone sensors due to the ubiquity of smartphones, and investigate the performance of different combinations of sensors for recognizing locomotion activities.

B. Deep Learning for Feature Learning

After collecting data from wearable sensors or an infrastructure, different activities can be recognized by applying different machine learning methods, such as artificial neural network [23], logistic regression algorithm [24], combination of neural networks and hidden Markov models [25], decision trees (DTs) [26], support vector machines (SVMs) [27], and conditional random fields [28]. While these conventional classifiers can be used to recognize different activities, their performance relies mainly on the suitability of handcrafted features of data. The design of these features requires expert knowledge and is a difficult and laborious task. For example, to achieve a satisfactory accuracy in LAR, one must consider extracting features of different types, such as statistical features (e.g., mean, variance, range, maximum, and minimum), frequency-domain features (e.g., Fourier transform), and time-domain features (e.g., autocorrelation coefficients). After the onerous task of feature extraction, a proper feature selection method [e.g., correlation-based feature selection (CFS) and sequential forward selection] is essential to guarantee an acceptable recognition performance. The feature extraction and selection processes not only involve laborious human intervention but also poorly generalize to other problem domains.

Deep learning methods have been proposed to extract features of data automatically, and have been applied in different application domains, such as image classification [12], natural language processing and speech recognition [11], and playing games [10]. The commonly used deep learning methods include stacked autoencoders [15], deep belief networks [31], convolutional neural networks [12], and recurrent neural networks [33]. These methods are originally proposed for

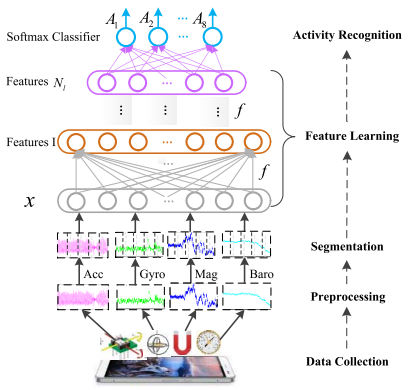


Fig. 1. Architecture of the proposed method.

TABLE I
TYPES OF LOCOMOTION ACTIVITIES

No.	Activity	Definition
A1	Still	The user remains still and does not use the phone.
A2	Walking	The user walks naturally with a phone.
A3	False Motion	The user remains still while using the phone for texting, calling, playing games, etc.
A4	Running	The user runs with the phone swinging in hand naturally.
A5	Upstairs	Going up stairs.
A6	Downstairs	Going down stairs.
A7	UpElevator	Taking an elevator upward.
A8	DownElevator	Taking an elevator downward.

image classification, natural language processing and speech recognition, but they can also be applied to human activity recognition. Indeed, deep learning methods have been used for activity recognition in a few recent works [34], [35], [37]. In this paper, we investigate the application of stacked denoising autoencoders for LAR based on four types of smartphone sensor data.

III. System Design

A. Architecture

The architecture of the proposed method for LAR with deep learning is shown in Fig. 1. It consists of data collection, preprocessing, segmentation, feature learning, and activity recognition steps. In the data collection step, accelerometer readings, gyroscope readings, magnetometer readings, and barometer readings together with their respective timestamps are recorded. In the following, we first introduce the locomotion activities of interest, and then elaborate the key steps of our method.

B. Locomotion Activities

We define eight types of locomotion activities that are critically important for indoor localization methods such as pedestrian dead reckoning, namely Still, Walking, Running, False Motion, Upstairs, Downstairs, UpElevator, and DownElevator, as shown in Table I. Seven of these activities are used in our previous work [6], but we add one new activity called False Motion, which means that the user remains still while using her phone for texting, watching movies, playing games, and so on. These activities have an important influence on the estimation of pedestrian step length and the detection of landmarks within indoor localization methods.

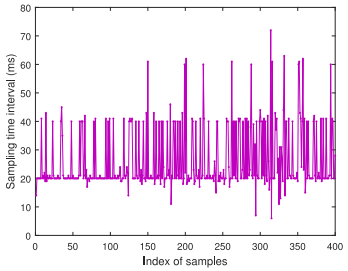


Fig. 2. Instability of sampling time interval.

C. Preprocessing

To combine data from different sensors for activity recognition, we need to align these sensor data according to their respective timestamps. It is observed that the sampling rate for the same sensor is not stable, as shown in Fig. 2. The time interval between the two neighboring samples of the accelerometer fluctuates from 7 to 72 ms, and it is the same case for the gyroscope, magnetometer, and barometer. In order to make the number of samples within a fixed time window stable, we use the *spline interpolation* to generate samples with a fixed sampling time interval. After this interpolation, we obtain 64 samples for each sequence of the accelerometer readings, gyroscope readings, and magnetometer readings, and 32 samples for the sequence of the barometer readings. The reason why the size of the sequence of barometer readings is smaller than the other sequences is that its sampling rate is much lower when set in the same sampling level (we set the sampling level in the Android app as the *SENSOR_DELAY_GAME* level for all the four sensors).

To reduce the correlation between adjacent sensor readings, we conduct the zero components analysis whitening operation [38] on the readings from the accelerometer, gyroscope, magnetometer, and barometer. This is to discover interesting regularities in the data by forcing the model to focus on higher order correlations [39].

D. Segmentation

Activity recognition cannot be done by using only a single data point as it cannot reflect the movement of a user. Therefore, we partition the sensor readings into sequences or segments according to a certain time frame. We consider four types of sensor readings in this paper, including readings from the accelerometer, gyroscope, magnetometer, and barometer. From the accelerometer readings, we can obtain three sequences $\{\boldsymbol{s}_i^{\mathbf{acc}_x}, \boldsymbol{s}_i^{\mathbf{acc}_y}, \boldsymbol{s}_i^{\mathbf{acc}_z}\}$. Each sequence corresponds to the readings along one axis. These sequences are created using a sliding window as follows:

$$\boldsymbol{s}_i^{\mathrm{acc}_x} = \left[\mathrm{acc}_t^x, \mathrm{acc}_{t+1}^x, \ldots, \mathrm{acc}_{t+K-1}^x\right] \quad (1)$$

$$\boldsymbol{s}_i^{\mathrm{acc}_y} = \left[\mathrm{acc}_t^y, \mathrm{acc}_{t+1}^y, \ldots, \mathrm{acc}_{t+K-1}^y\right] \quad (2)$$

$$\boldsymbol{s}_i^{\mathrm{acc}_z} = \left[\mathrm{acc}_t^z, \mathrm{acc}_{t+1}^z, \ldots, \mathrm{acc}_{t+K-1}^z\right] \quad (3)$$

where K is the sequence size, which is 64 in this paper (corresponding to 2 s). It should be noted that the value of K is important for accurately recognizing different activities. A too small value may not precisely capture the full characteristics of the activity, while a too large sequence may include more than one activity. Similarly, we can obtain sequences $\{\boldsymbol{s}_i^{\text{gyro}_x}, \boldsymbol{s}_i^{\text{gyro}_y}, \boldsymbol{s}_i^{\text{gyro}_z}\}$ from the gyroscope readings, $\{\boldsymbol{s}_i^{\text{mag}_x}, \boldsymbol{s}_i^{\text{mag}_y}, \boldsymbol{s}_i^{\text{mag}_z}\}$ from the magnetometer readings, and $\{\boldsymbol{s}_i^{\text{pres}}\}$ from the barometer readings.

E. Deep Learning for Locomotion Activity Recognition

In this part, we describe the proposed method based on the stacked denoising autoencoders [8]. The denoising autoencoder is an extension of the autoencoder [40], [41], which can discover more robust features and prevent it from simply learning similarly uninteresting ones. A denoising autoencoder is trained to reconstruct the original input from a corrupted version of it. Let $\boldsymbol{x}_i$ represent the input vector at time i, which consists of partitioned sequences of sensor readings, namely

$$\boldsymbol{x}_i = \Big[\boldsymbol{s}_i^{\text{acc}_x}, \boldsymbol{s}_i^{\text{acc}_y}, \boldsymbol{s}_i^{\text{acc}_z}, \boldsymbol{s}_i^{\text{gyro}_x}, \boldsymbol{s}_i^{\text{gyro}_y} \boldsymbol{s}_i^{\text{gyro}_z}, \boldsymbol{s}_i^{\text{mag}_x}, \boldsymbol{s}_i^{\text{mag}_y}, \boldsymbol{s}_i^{\text{mag}_z}, \boldsymbol{s}_i^{\text{pres}}\Big]^T \tag{4}$$

where $\boldsymbol{x}_i$ is an $M \times 1$ vector (M equals to 608 in this paper). The size of the sequence of $\boldsymbol{s}_i^{\text{pres}}$ is 32 while the size of other sequences is 64. Each element of $\boldsymbol{x}_i$ corresponds to an input unit in the input layer. These input units are then corrupted into $\boldsymbol{x}_i'$ by using a stochastic mapping q, namely,

$$\boldsymbol{x}_i' = q(\boldsymbol{x}_i). \tag{5}$$

In this paper, we do the corruption operation by adding the masking noise [8], which involves forcing a fraction ν of $\boldsymbol{x}_i$ to be 0 (we set ν to 0.5 in this paper).

The proposed method uses stacked denoising autoencoders for LAR, which consists of multiple layers of denoising autoencoders in which the output of each layer is used as the input of the successive layer. We first explain the principle of a denoising autoencoder. It has two processes: 1) encoding and 2) decoding. In the encoding process, the input vector is transformed into features in a hidden layer, which can be reconstructed to approximate the input in the decoding processing. The learning process is done by minimizing the reconstruction error between the input data and its reconstruction. The encoding is done by applying a sigmoid function f to the input

$$\boldsymbol{a} = f\big(\boldsymbol{W}_1\boldsymbol{x}_i' + \boldsymbol{b}_1\big) \tag{6}$$

where $\boldsymbol{W}_1$ is an $N \times M$ encoding matrix, $\boldsymbol{a}$ and $\boldsymbol{b}_1$ are N-dimensional activation vector and bias vector, respectively. N is the number of units in the hidden layer, and M is the number of input units. The decoding is done by conducting a similar process, namely,

$$\widehat{\boldsymbol{x}}_i = g(\boldsymbol{W}_2\boldsymbol{a} + \boldsymbol{b}_2) \tag{7}$$

where $\boldsymbol{W}_2$ is an $M \times N$ decoding matrix, and $\boldsymbol{b}_2$ is an M-dimensional bias vector. $\widehat{\boldsymbol{x}}_i$ is the reconstructed vector of the input vector $\boldsymbol{x}_i$ and g is also a sigmoid function. The minimization of the reconstruction error can be done by minimizing the square error loss function $L(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}_i)$, where

$$L(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}_i) = \frac{1}{2}\sum_{j=1}^{M}\big\|x_j - \widehat{x_j}\big\|^2. \tag{8}$$

It should be noted that the error computation in the denoising autoencoder is exactly the same as in the autoencoder. In other words, the error computation uses the original input rather than the corrupted version.

To enable the stacked denoising autoencoders to work even when the number of hidden units is large (e.g., the default number of hidden units is 1000, which is larger than 608—the number of input units), we apply a sparsity constraint to the activation function. Therefore, the cost function is written as

$$J_{\text{dae}} = L(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}_i) + \beta\sum_{j=1}^{N} KL\Big(\rho\Big\|\hat{\rho}_j\Big) \tag{9}$$

where $KL(\rho||\hat{\rho}_j) = \rho\log(\rho/\hat{\rho}_j) + (1-\rho)\log[(1-\rho)/(1-\hat{\rho}_j)]$ is the Kullback–Leiber divergence [42] between the sparsity parameter ρ (we set $\rho = 0.05$) and the average activation $\hat{\rho}_j$ of hidden unit j. β is the sparsity penalty (which is set to 1 in this paper), and N is the number of hidden units.

The stacked denoising autoencoders learns features in an unsupervised manner by using the greedy layer-wise training method [43]. The training process of the stacked denoising autoencoders starts by training the first layer on the input to learn the features in the first hidden layer. Similarly, this operation is repeatedly conducted for subsequent layers to finish the whole training process. After the layer-wise training, a fine-tuning operation is followed to optimize the parameters of all layers through backpropagation, which will further improve the results.

At the top of this network is a softmax classifier, which is used to classify the activities of interest. These activities are denoted by A_j ($j = 1, 2, \ldots, 8$). Then, the activity recognition by the softmax classifier can be written as

$$p\big(y = A_j|\boldsymbol{x}_i; \boldsymbol{\theta}\big) = \frac{e^{\boldsymbol{\theta}_j^T\boldsymbol{x}_i}}{\sum_{k=1}^{8} e^{\boldsymbol{\theta}_k^T\boldsymbol{x}_i}} \tag{10}$$

where $\boldsymbol{\theta}$ is the parameter vector of the network.

The pseudo-code for the LAR using the stacked denoising autoencoders is described in Algorithm 1. It starts with data preprocessing and segmentation. Then the network is trained to learn useful features in a layer-wise manner. The feature learning process is conducted on the unlabeled training dataset, which means that there is no need for label information of activities. After the layer-wise training, a fine-tuning operation using labeled dataset is followed to optimize the parameters of all layers through backpropagation, which will improve the results. Note that the corruption operation is only used for the initial denoising training of each individual layer to learn the parameters of the network. However, there is no corruption for producing the output that will serve as the input of the next layer.

Algorithm 1: Deep LAR Method

Input : Unlabeled training dataset $D_{unlabeled} = \{X_i^u\}$, labeled training dataset $D_{labeled} = \{X_i^{tr}, Y_i^{tr}\}$, unlabeled testing dataset $D_{test} = \{X_i^{te}\}$

Output: Activity labels $\boldsymbol{A}_j$ of the unlabeled testing data

// Initialization:
Initialize the parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ for each layer
Stabilize the number of samples witin a certain time period for each type of sensor readings using the spline interpolation
Apply the zero components analysis whitening to each type of sensor readings
Segment the whitened data into sequences, and obtain the input vector $\boldsymbol{x}_i$
// Unsupervised training on unlabeled training dataset $D_{unlabeled}$:
repeat
  Corrupt the input vector by adding masking noise and obtain $\boldsymbol{x}_i'$
  Train the l-th layer of the stacked denoising autoencoders using the corrupted data sequences, and obtain the network parameters $\boldsymbol{\theta}_l$
  Compute the output of the l-th layer by using the learnt parameters $\boldsymbol{\theta}_{1:l}$ on the uncorrupted input, which will feed to the $l+1$-th layer as input
until $l++ == N_l$;
Use labeled dataset $D_{labeled}$ to train the top layer by the softmax classifier
Fine-tune the entire network through backpropagation
// Testing:
Use the trained network to predict the labels A of dataset D_{test}

IV. Experiments and Results

A. Experimental Setup

The proposed LAR method was evaluated by a series of experiments. Twelve volunteers were recruited to conduct these activities defined in Table I with a Samsung Galaxy S III phone in hand. Table III shows the height and gender of the participants. All participants were healthy, and aged between 25 and 35. In this paper, we consider only the case that the phone is carried in the hand. Readers who are interested in the effect of different poses on the classification accuracy are referred to [6] and [44].

During the experiments, the ground truth labels were recorded by an app we developed for this research. Before conducting an activity, the participant was asked to select the activity on the app to record the corresponding ground truth labels. The data we collected included readings from the accelerometer, gyroscope, magnetometer, and barometer. To precisely evaluate the accuracy of our LAR method, we manually modified the labels that were mistakenly selected during experiments and removed noise between the transition of two activities (e.g., from taking an elevator to walking). The data size is shown in Table II. After the spline interpolation, the dimension of each input sequence for 2 s time interval is 608 for the combination of four types of sensor readings.

TABLE II
Size of Input Data (3 Indicates Consideration for Three Axes)

Sensors	Number of samples	Number of segments after interpolation (for 2s time interval)
Acc	3× 902026	3 × 12822
Gyro	3× 995973	3 × 12822
Magnet	3× 712073	3 × 12822
Baro	1× 497964	1 × 12822

TABLE III
User Profile

User No.	Height (cm)	Gender	User No.	Height (cm)	Gender
1	166	male	7	163	female
2	173	male	8	165	female
3	178	male	9	164	female
4	158	female	10	166	female
5	170	male	11	178	male
6	162	female	12	170	male

TABLE IV
List of Hyperparameters for Deep Networks

Hyperparameter	Description	Considered values
N_l	number of hidden layers	{**2**,3,4,6}
N_h	number of units per hidden layer (same for all layers)	{200, 500, **1000**, 1500, 2000 }
N_{epoch}	number of pretraining epochs	{100, **200**, 300}
ν	corrupting noise level	{0.1, 0.3, **0.5**, 0.7}
T	segment size (in seconds)	{1, **2**, 4, 6}
α	learning rate	{1×10^{-4}, 5×10^{-4}, $\mathbf{1}\times10^{-3}$, 5×10^{-3} }
β	weight of sparsity penalty term	{**1**, 2, 3, 5}

B. Hyperparameter Settings

Table IV gives a list of the hyperparameters we considered in this paper. To reduce the selection space, we let all the hidden layers share the same number of units, the same learning rate, and the same noise level. It should be noted that the bold value for each hyperparameter is used in the following analysis when there is no mention specifically.

C. Classification Performance

The evaluation metrics we used were *precision*, *recall*, and *F-measure* [45]. The *precision* for an activity is the number of correctly labeled segments (true positives) divided by the total number of segments labeled as belonging to this activity (the sum of true positives and false positives). The *recall* is defined as the number of correctly labeled segments divided by the total number of segments that actually belong to this activity (which is the sum of true positives and false negatives). The *F-measure* is a measure of testing accuracy that considers both the precision and the recall.

The fivefold cross validation method was used to verify the performance of the proposed method. All data sequences were randomly divided into five groups of the same size, where one group was retained as the validation data for testing the model and the remaining four were used as the training data. The cross validation process was repeated five times with each of these five groups used exactly once as the testing data.

We first analyze the performance of the proposed method for recognizing each activity, which is shown in

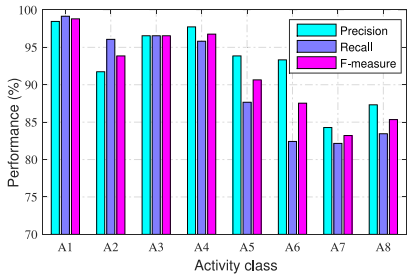


Fig. 3. Recognition performance for each activity. A network of two layers and 1000 neurons for each layer is used. The input vector consists of the readings from the accelerometer, gyroscope, magnetometer, and barometer.

TABLE V
CONFUSION MATRIX OF TESTING (TWO LAYERS, 1000 NEURONS FOR EACH LAYER)

Ground Truths \ Testing Results	A1	A2	A3	A4	A5	A6	A7	A8
A1	**2116**	9	1	0	1	5	0	2
A2	13	**4374**	41	10	32	30	30	23
A3	11	45	**2349**	1	14	2	8	3
A4	0	21	3	**733**	3	5	0	0
A5	5	115	8	1	**1023**	11	1	3
A6	4	130	17	0	12	**769**	0	1
A7	0	36	10	3	2	0	**327**	20
A8	0	38	4	2	3	2	22	**358**

Fig. 3 and Table V. Generally, the locomotion activities involved with vertical movement (A5-Upstairs, A6-Downstairs, A7-UpElevator, and A8-DownElevator) have poorer performance than the horizontal locomotion activities (A1-Still, A2-Walking, A3-False Motion, and A4-Running). This is because vertical locomotion activities are more likely to be misrecognized as other activities. As can be seen in Table V, 115 out of 1167 segments of Upstairs are incorrectly classified as Walking, and 130 out of 933 segments of Downstairs are wrongly recognized as Walking. The main reason for this misrecognition is that Upstairs and Downstairs share similar characteristics as Walking. UpElevators and DownElevators are also easily misrecognized as Walking since the users may not remain still when taking an elevator. Another possible reason for other activities being misrecognized as Walking is that the training dataset is unbalanced. Walking activity has more samples than other activities, and thus the classification results are biased toward Walking activity. The overall F-measure accuracy of the proposed method using two layers with 1000 neurons per layer is 94.04%. In the following, we use F-measure as the performance metric to evaluate the effect of different settings and parameters.

D. Combination of Different Sensors

While most existing research works use only the accelerometer readings for activity recognition, we investigate the performance of activity recognition algorithms using different types of sensors and their combinations (Table VI). In general, the more types of sensor readings combined, the better the recognition. Specifically, combining the accelerometer readings, gyroscope readings, magnetometer readings, and the barometer readings achieves the best accuracy (94.04% as measured by F-measure). The combination of the accelerometer, magnetometer, and barometer outperforms other combinations when using only three types of sensors. The integration

TABLE VI
PERFORMANCE COMPARISON OF USING DIFFERENT TYPES OF SENSORS

Number of sensors	Sensors	F-measure (%)
1	Acc	**88.83**
	Gyro	79.83
	Baro	52.16
	Magnet	80.15
2	Acc + Gyro	89.81
	Acc + Baro	91.66
	Acc + Magnet	**92.46**
	Gyro + Baro	85.39
	Gyro + Magnet	88.58
	Magnet + Baro	87.28
3	Acc + Gyro + Baro	92.58
	Acc + Magnet + Baro	**93.57**
	Acc + Gyro + Magnet	92.84
	Gyro + Magnet + Baro	91.71
4	Acc + Gyro + Magnet + Baro	**94.04**

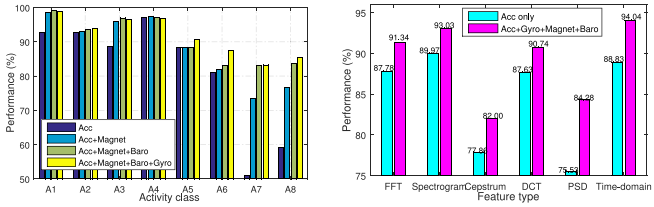


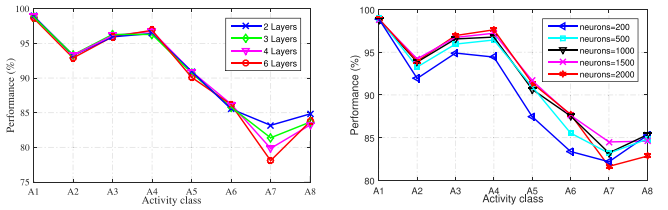
Fig. 4. (a) Per-activity recognition performance for different combinations of sensors. (b) Performance comparison of using frequency-domain features and time-domain features (two layers, 1000 neurons for each layer).

of the accelerometer and the magnetometer outperforms other combinations of only two types of sensors. When considering individual sensors, the accelerometer achieves a better recognition performance than the other sensors. In the following, we use the combination of four types of sensors for analyzing the performance of different network settings and the influence of different parameters.

It is also interesting to see how the addition of a type of sensors affects the classification accuracy of each activity. Therefore, we compare the performance of each of the best combinations for each activity, as shown in Fig. 4(a). The combination of the accelerometer and the magnetometer can significantly improve the classification for Still (A1), Running (A3), UpElevator (A7), and DownElevator (A8) compared to the results using only the accelerometer. The addition of the barometer further enhances the accuracy for UpElevator and DownElevator. The introduction of the gyroscope increases the recognition rate for Upstairs (A5) and Downstairs (A6). Overall, the combination of all four sensors achieves the best accuracy on average.

E. Frequency-Domain Feature Analysis

The transformation from raw time-domain data to frequency-domain data has been shown in the literature to improve the activity recognition accuracy [46]. We consider preprocessing raw data using the fast Fourier transform, spectrogram analysis, cepstrum analysis, power spectral density, and discrete cosine transform, respectively. We compare the results of conducting these transforms on the accelerometer data only and on the combination of four types of sensors (accelerometer + gyroscope + magnetometer + barometer).



(a) (b)

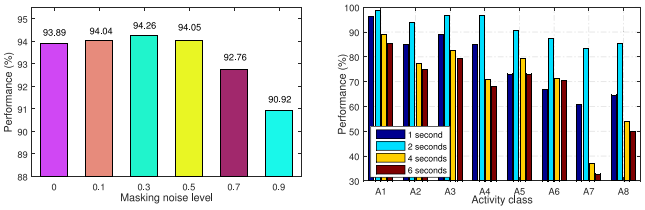
Fig. 5. (a) Influence of using different number of layers on the recognition performance per activity (500 neurons per layer). (b) Influence of using different number of neurons per layer on the recognition performance per activity.

TABLE VII
PERFORMANCE COMPARISON OF USING DIFFERENT NUMBER OF LAYERS (500 NEURONS FOR EACH LAYER)

Number of layers	F-measure (%)	
	500 neurons	1000 neurons
2	93.60	94.04
3	93.55	93.93
4	93.53	94.07
6	93.19	93.81

TABLE VIII
PERFORMANCE COMPARISON OF USING DIFFERENT NUMBER OF NEURONS (TWO LAYERS)

Number of neurons per layer	F-measure (%)
200	92.32
500	93.60
1000	94.04
1500	94.34
2000	94.16



(a) (b)

Fig. 6. (a) Influence of masking noise levels on the overall recognition (two layers, 1000 neurons per layer). (b) Influence of segment size on the recognition performance per activity.

As shown in Fig. 4(b), the best accuracy is achieved by using spectrogram coefficients as input when using only the accelerometer readings. However, when the input consists of readings from all four sensors, our results do not support the previous findings that frequency-domain features lead to superior recognition performance. On the contrary, the recognition accuracy obtained by time-domain features is better than the accuracy of all frequency-domain features. It also demonstrates that the recognition using the combination of sensors outperforms that using the accelerometer only in all cases.

F. Sensitivity of Parameters

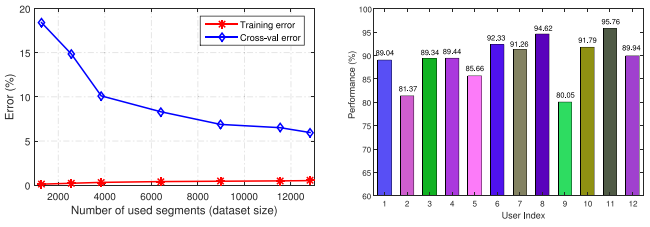
In this section, we analyze the sensitivity of important parameters, including the number of layers, the number of neurons per layer, masking noise level, segment size, and training completeness. Other parameters such as the learning rate α, the weight of sparsity penalty term β, and the number of pretraining epochs are simply set to the default values as shown in Table IV, which are empirically determined.

1) Number of Layers: Fig. 5(a) and Table VII show the classification accuracy of each activity by using different number of layers. It turns out that the number of layers does not have a significant influence on the performance of LAR. Having more layers does not necessarily increase the performance, but would raise the computational cost. This may be because the data segments of some activities are not sufficient to train a complex network with more layers.

2) Number of Neurons: Table VIII shows the influence of the number of neurons on the classification performance. The increase of the number of neurons improves the classification accuracy. Specifically, a performance improvement of 1.28% is achieved by increasing the number of neurons from 200 to 500. The increase trend continues until the number of neurons reaches 1500 at which point increasing the number of neurons does not further improve the accuracy. The influence of the number of neurons on the recognition performance for each activity is shown in Fig. 5(b), which also demonstrates the increase of accuracy over the number of neurons. It should be pointed out that the activities A7 (UpElevator) and A8 (UpElevator) witness a decrease when raising the number of neurons. This is probably because the number of training samples for A7 and A8 is not sufficient to adequately train a complex network with many neurons (see Table V). Usually, including more neurons means higher computational cost and larger memory requirement. There is usually a tradeoff between the classification accuracy and the computational cost.

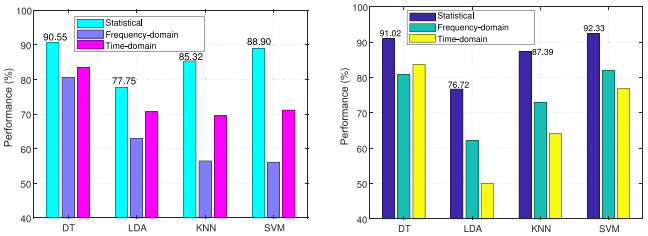
3) Masking Noise Level: The masking noise level indicates the fraction of elements of the input data that are forced to 0. Fig. 6(a) demonstrates the influence of different masking noise levels on the classification accuracy. The best performance arises at around the corruption ratio (or masking noise level) of 0.3, from which increasing or decreasing the ratio will lead to a decrease in the classification accuracy. Note that when the corruption ratio equals 0, the model becomes a stacked sparse autoencoder.

4) Segment Size: To analyze the effect of segment size, the classification was performed with segments of different size. As shown in Fig. 6(b), the best performance is achieved when using 2 s time interval as the segment size. Decreasing the time interval to 1 s diminishes the per-activity recognition rate. This is because a small time interval may not capture the movement characteristics of users' locomotion activities. It is also interesting to see the decrease in the performance when increasing the interval from 2 to 4 s, and then to 6 s. The major reason is that increasing the segment size will decrease the number of samples. For example, the number of samples segmented with 4 s interval halves that segmented with 2 s interval. Another possible reason is that a large segment size may include data from multiple activities.



(a) (b)

Fig. 7. (a) Learning curve of stacked denoising autoencoders. (b) Generalization capability of the proposed LAR model on unknown subject.



(a) (b)

Fig. 8. (a) Performance of the benchmark classifiers without feature selection. (b) Performance of the benchmark classifiers with the CFS method.

5) Adequacy of Training: We use the learning curve to verify whether our LAR model is sufficiently trained. Fig. 7(a) shows that the cross-validation (testing) error decreases with the increase of the number of dataset size while the training error remains almost unchanged. The testing error decreases dramatically when the number of used segments raises from about 1000 to near 4000, after which it decreases gradually. It is expected that the testing error will further decrease with more data available, but this decrease will not be significant. Note that when we use a small subset of data for training and testing, the training error and testing error are calculated using this subset, rather than the full data set.

6) Generalization Capability on Unknown Subject: To analyze the generalization capability of the proposed model on unknown subject, the *leave-one-subject-out* test is used. The data from each user are in turn used as the testing data, while the remaining eleven users' data are used as the training data. This process was repeated 12 times with each user's data being used exactly once as the testing data. The results are shown in Fig. 7(b), from which we can see that the performance of the proposed model varies from user to user, ranging from 80.05% to 95.76%. The major reason for this is that different users have different movement characteristics. The recognition rate for user 2 and user 9 is relatively poor, probably because their activities share less common features with other users.

G. Comparison With Conventional Machine Learning Methods

We compare the proposed deep LAR method with the commonly used conventional supervised machine learning methods, including DTs, linear discriminant analysis (LDA), K-nearest neighbors, and SVMs. In order to use these conventional classifiers, we first need to define and extract features. In this paper, we extracted statistical features (mean, maximum, minimum, variance, and range), frequency-domain features (the first three most dominant frequencies and energies of discrete Fourier transform), and time-domain features (autocorrelation coefficients and cross correlation coefficients) from accelerometer readings, gyroscope readings, and magnetometer readings. The pressure difference computed from barometer readings was also fed to these classifiers. In total, 136 features were extracted, including 45 features from each of the accelerometer, gyroscope, and magnetometer data, and 1 feature from the barometer readings.

We compare the results of the four classifiers using the CFS method [47] with those without using the feature selection method, as shown in Fig. 8(a) and (b). We can see that the performance of the four classifiers changes significantly when using different set of features. Generally, using statistical features can achieve better performance than using frequency-domain features and using time-domain features. The use of the CFS method can help to achieve a better recognition rate except for LDA since LDA usually requires to take more features as input to achieve a relatively high accuracy.

Overall, the proposed activity recognition method performs better than the benchmark classifiers. More importantly, it does not require manual feature design. The performance of conventional classifiers relies on the features extracted and selected, which may vary significantly from feature to feature.

V. Conclusion

This paper presents a deep learning method for recognizing different locomotion activities that are relevant to indoor localization and navigation. The proposed method based on stacked denoising autoencoders allows to learn features of data automatically, eliminating the need to manually design features. Consequently, the proposed method is independent of expert knowledge and significantly reduces the effort for manual feature design. Experimental results show that the proposed method achieves a higher accuracy than other classifiers. We also show that using the combination of data from multiple sensors can achieve a better activity recognition performance than using the acceleration data only.

References

[1] F. Gu, K. Khoshelham, and S. Valaee, "Locomotion activity recognition-a deep learning approach," in *Proc. 28th Annu. IEEE Int. Symp. Personal Indoor Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, 2017, pp. 1–5.

[2] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[3] S. Wang and G. Zhou, "A review on radio based activity recognition," *Digit. Commun. Netw.*, vol. 1, no. 1, pp. 20–29, 2015.

[4] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[5] O. D. Incel, M. Kose, and C. Ersoy, "A review and taxonomy of activity recognition on mobile phones," *J. Bionanosci.*, vol. 3, no. 2, pp. 145–171, 2013.

[6] F. Gu, A. Kealy, K. Khoshelham, and J. Shang, "User-independent motion state recognition using smartphone sensors," *Sensors*, vol. 15, no. 12, pp. 30636–30652, 2015.

[7] J. Li *et al.*, "Feature selection: A data perspective," *ACM Comput. Surveys*, vol. 50, no. 6, p. 94, 2017.
[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.
[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[10] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
[11] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[13] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3995–4001.
[14] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.
[15] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.
[16] F. Zhu, L. Shao, J. Xie, and Y. Fang, "From handcrafted to learned representations for human action recognition: A survey," *Image Vis. Comput.*, vol. 55, pp. 42–52, Nov. 2016.
[17] T. Sohn *et al.*, "Mobility detection using everyday GSM traces," in *Proc. 8th Int. Conf. Ubiquitous Comput.*, 2006, pp. 212–224.
[18] S. Sigg, U. Blanke, and G. Troster, "The telepathic phone: Frictionless activity recognition from WiFi-RSSI," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, 2014, pp. 148–155.
[19] Y. Wang *et al.*, "E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures," in *Proc. ACM 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 617–628.
[20] Y. Gu *et al.*, "MoSense: An RF-based motion detection system via off-the-shelf WiFi devices," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2326–2341, Dec. 2017.
[21] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.
[22] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.
[23] M.-W. Lee, A. M. Khan, and T.-S. Kim, "A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation," *Pers. Ubiquitous Comput.*, vol. 15, no. 8, pp. 887–898, 2011.
[24] Ó. D. Lara, A. J. Pérez, M. A. Labrador, and J. D. Posada, "Centinela: A human activity recognition system based on acceleration and vital sign data," *Pervasive Mobile Comput.*, vol. 8, no. 5, pp. 717–729, 2012.
[25] C. Zhu and W. Sheng, "Motion- and location-based online human daily activity recognition," *Pervasive Mobile Comput.*, vol. 7, no. 2, pp. 256–269, 2011.
[26] I. Kouris and D. Koutsouris, "A comparative study of pattern recognition classifiers to predict physical activities using smartphones and wearable body sensors," *Technol. Health Care*, vol. 20, no. 4, pp. 263–275, 2012.
[27] J.-L. Reyes-Ortiz, L. Oneto, A. Samá, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, Jan. 2016.
[28] D. H. Hu, S. J. Pan, V. W. Zheng, N. N. Liu, and Q. Yang, "Real world activity recognition with multiple goals," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, Seoul, South Korea, 2008, pp. 30–39.
[29] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[30] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3104–3112.
[31] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 4, pp. 778–784, Apr. 2014.
[32] T. N. Sainath *et al.*, "Deep convolutional neural networks for large-scale speech tasks," *Neural Netw.*, vol. 64, pp. 39–48, Apr. 2015.
[33] F. J. Ordonez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
[34] C. A. Ronao and S. B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *Proc. Int. Conf. Neural Inf. Process.*, 2015, pp. 46–53.
[35] X. Li, Y. Zhang, I. Marsic, A. Sarcevic, and R. S. Burd, "Deep learning for RFID-Based activity recognition," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2016, pp. 164–175.
[36] L. Wang, "Recognition of human activities using continuous autoencoders with wearable sensors," *Sensors*, vol. 16, no. 2, p. 189, 2016.
[37] V. Radu *et al.*, "Towards multimodal deep learning for activity recognition on mobile devices," in *Proc. ACM Adjunct Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 185–188.
[38] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.
[39] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.
[40] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[41] A. Y. Ng, J. Ngiam, C. Y. Foo, Y. Mai, and C. Suen. *Sparse Autoencoder/Preprocessing: PCA and Whitening*. Accessed: Nov. 1, 2016. [Online]. Available: http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial
[42] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.
[43] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 153–160.
[44] D. Roggen, K. Förster, A. Calatroni, A. Bulling, and G. Tröster, "On the issue of variability in labels and sensor configurations in activity recognition systems," in *Proc. Workshop 8th Int. Conf. Pervasive Comput. Best Pract. Activ. Recognit.*, 2010, pp. 1–4.
[45] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[46] M. A. Alsheikh *et al.*, "Deep activity recognition models with triaxial accelerometers," in *Proc. Workshops 30th Conf. Artif. Intell. Appl. Assistive Technol. Smart Environ. Tech. Rep. (AAAI)*, 2016, pp. 8–13.
[47] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 856–863.

Authors' photographs and biographies not available at the time of publication.