Multimed Tools Appl
DOI 10.1007/s11042-017-5100-4

Latent feature learning for activity recognition using simple sensors in smart homes

Guilin Chen[1] · **Aiguo Wang**[2] · **Shenghui Zhao**[1] · **Li Liu**[3] · **Chih-Yung Chang**[4]

Received: 12 February 2017 / Revised: 30 June 2017 / Accepted: 9 August 2017

Abstract Activity recognition is an important step towards monitoring and evaluating the functional health of an individual, and it potentially promotes human-centric ubiquitous applications in smart homes particularly for senior healthcare. The nature of human activity characterized by a high degree of complexity and uncertainty, however, poses a great challenge to the design of good feature representations and the optimization of classifiers towards building a robust model for human activity recognition. In this study, we propose to exploit deep learning techniques to automatically learn high-level features from the binary sensor data under the assumption that there exist discriminative latent patterns inherent in the simple low-level features. Specifically, we extract high-level features with a stacked autoencoder that has a deep and hierarchy architecture, and combine feature learning and classifier construction into a unified framework to obtain a jointly optimized activity recognizer. Besides, we investigate two different original feature representations of the sensor data for latent feature learning. To evaluate the performance of the proposed method, we conduct extensive experiments on three publicly available smart home datasets, and compare it with a range of shallow models in terms of time-slice accuracy and class accuracy. Experimental results show that our proposed model achieves better recognition rates and generalizes better across different original feature representations, indicating its applicability to the real-world activity recognition.

Keywords Activity recognition · Smart home · Feature learning · Autoencoder · Shallow model

✉ Aiguo Wang
wangaiguo2546@163.com

[1] School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China

[2] School of Computer and Information, Hefei University of Technology, Hefei 230009, China

[3] School of Software Engineering, Chongqing University, Chongqing 400044, China

[4] Department of Computer Science and Information Engineering, Tamkang University, Taipei 25157, Taiwan

Published online: 18 August 2017

1 Introduction

The rapid development of pervasive computing technology along with the minimization of sensors make it possible for us to customize and provide ubiquitous and context-aware services to individuals living in a smart home [23]. On the other hand, due to the ever-increasing global aging population, the high expenditure of healthcare costs as well as the growing desire of subjects to remain independent in their own home, ambient assisted living (AAL) systems, which can perceive and act on physical surroundings using different types of sensors and recognize human activities of daily living (ADLs), are needed [6]. In such systems, accurately recognizing human activities (e.g., cooking, eating, drinking, grooming, washing, and sleeping) is an important step towards evaluating the functional ability of the residents for independent living [15]. Besides, reliably recognizing on-going activities in a home setting potentially facilitates a large number of meaningful applications that range from fall detection, behavior analysis, and activity reminder to chronic disease management and rehabilitation feedback [18, 22].

Activity recognition (AR) is a meaningful yet challenging research topic. To obtain high recognition rates and adapt to different application scenarios, researchers have explored a variety of sensing technologies and methods. Generally, they can be broadly grouped into three categories: vision-based techniques (e.g., camera and video), wearable sensor-based techniques (e.g., accelerometer, gyroscope, smartphone, and RFID reader), and environment interactive sensor-based techniques (e.g., motion detector, pressure sensor, and contact sensor) [13]. In contrast to vision-based and wearable sensor-based techniques, environment interactive sensor-based approaches have advantages of low costs, easy deployment, and inherent non-intrusiveness. Therefore, they are considered a promising way to monitor human physical activities and evaluate cognitive health especially when privacy and user acceptance issues are considered [23]. In practice, environment interactive sensor-based approaches infer the ADLs that are performed by an individual via capturing the interactions between the individual and a specific object. For example, we can use a contact sensor to record whether and when a medication container is used for the application of adherence to medication.

From the view of machine learning, the input of an activity recognition system is a stream of sensor activations [19]. Herein, we can treat sensor-based activity recognition as a time series analysis problem, and our aim is to associate a continuous portion of streaming sensor data with one of the pre-defined known activities. One of the commonly-used methods to AR is the supervised learning that has an explicit training phase, and it mainly consists of three steps [27]. First, a stream of raw sensor data is divided into segments with the sliding window technique. Specifically, a window with a fixed time-length or a fixed number of sensor events is shifted along the stream with (non-) overlapping between adjacent segments. The next step is feature engineering and model training. That is, we extract various time-domain and frequency-domain features (e.g., max, min, mean, variance, and entropy) from each segment and represent the raw signals with the feature vector, and then train an activity recognizer with those feature vectors. The last phase, called activity recognition, is to use the trained classifier to classify a stream of sensor data. It has been known to us that a good feature representation of sensor data, suitable choices of a classifier and its parameter settings are three crucial factors that largely determine the performance of an AR system [20]. Although researchers have proposed many models to recognize human ADLs, the performance of most of existing methods heavily rely on hand-crafted features that are extracted from the time-domain and frequency-domain. Also, most classifiers used have shallow structures and they probably fail

to capture the underlying latent information hidden in the raw streaming sensor data [2]. In addition, in most studies, the extraction of features and the training of classifiers are treated as two separate steps, thus they are not jointly optimized. Consequently, the way to obtain a good feature descriptor is not clear without the proper guidance of classification performance, and we may be unable to obtain satisfactory performance without the exploration of feature extraction.

In recent years, deep learning techniques have gained great popularity in both academic and industrial areas, and they have been successfully applied in many fields such as speech recognition, face recognition, and image classification. Deep learning, also called unsupervised feature learning, enables us to automatically learn high-level features from the original low-level features without the help of specific domain knowledge but with a general-purpose learning procedure. In this study, with the aim to improve the recognition performance of an environment interactive sensor-based AR system, we plan to explore the power of deep learning techniques for AR in a smart home setting to discover the latent information that are inherent in simple sensors. Specifically, we use a stacked denoising autoencoder to learn a robust latent feature representation of the original features from a large amount of unlabeled data, and then we direct the learnt features as input of a top classifier. Accordingly, we can unify feature learning and classifier training in a single pipeline and jointly fine-tune parameter values using the labeled data. This enables us to obtain a robust activity recognition model that can automatically learn high-level features towards better recognition performance. The main contributions of this study are as follows. (1) We explore the application of deep learning techniques for human activity recognition in a smart home setting that is equipped with simple sensors. In contrast to the sensors with a fixed sampling rate, these simple sensors are trigged by the interaction between an individual and an object, and their firings have binary values. (2) We consider two different original feature representations of the raw sensor data, and experimentally evaluate their roles in deep learning models and shallow models. (3) Two different evaluation metrics, time-slice accuracy and class accuracy, are used to comprehensively evaluate the effectiveness of an activity recognizer. (4) Extensive experiments were conducted to verify the effectiveness of proposed method on three publicly available smart home datasets. The results show the superior of deep learning over its competitors that have a shallow structure. In addition, the results show that a model with two hidden layers obtains better recognition performance than a model with one hidden layer. This indicates that a deeper model probably contributes to the improvement of an AR system.

The rest of this paper is structured as follows. Section 2 briefly reviews related work in activity recognition. We then present the proposed deep learning based AR model and detail its core components in section 3. Experimental setup and results are presented and analyzed in section 4. The last section concludes this study with a summary and points out the possible future research lines.

2 Related work

To improve the performance of activity recognition and enable its wide applications in real world scenarios, researchers have done considerable work in exploring various sensing technologies and models [13]. It has been shown that different types of sensor modalities are suitable for different scenarios and have their own advantages of recognizing different activities. For vision-based methods, they generally achieve a better recognition rate, but the

use of camera or video is not practical in many indoor environments particularly when the privacy issue is considered [5]. Besides, vision-based methods face technical challenges arising from light, distance from cameras, occlusion and low object recognition rate, which also limit their wide use.

In the past few years, researchers have designed a variety of sensors and used them for human activity recognition [8]. These sensors can be categorized into wearable sensors and environment interactive sensors. Wearable sensors, such as accelerometer, gyroscope, GPS, and magnetometer, are worn or carried by an individual when performing an activity, and they generally generate different raw sensor signals for different activities. For example, Bao and Intille used five small biaxial accelerometers that were worn simultaneously on different parts of the body (four limb positions plus the right hip) to recognize twenty daily activities. By collecting experimental data from twenty volunteers and extracting time-domain and frequency-domain features, they compared the recognition rates of three different classifiers. The results showed that decision tree obtained the best performance with an accuracy of 84.0% [1]. Tapia et al. proposed to implement a real-time system that can recognize physical activities and corresponding intensities using a heart rate monitor and five tri-axial accelerometers placed on right arm, right leg, and the waist [24]. They then applied their system to recognize thirty physical gymnasium activities, where they obtained a 94.6% subject-dependent recognition rate, a subject-independent accuracy of 56.3%, and an accuracy of 80.6% without differentiating the activity intensities. Besides, RFID technology also provides us a way to recognize human activities, because they can capture the interaction between an individual and a specific object. For example, Kim et al. built a real-time indoor healthcare monitoring system to locate and track the elderly in the situation where someone wears a RFID reader and RFID tags are attached to the objects [12]. Also, Philipose et al. used RFID technology and a probabilistic engine for fine-grained activity recognition [19].

Nowadays, with the increasing power of mobile devices in processing and communication, smartphones are usually embedded with built-in accelerometers, gyroscopes, and GPS. Because no additional equipment is required for data collection and processing [14, 30], smartphones are a priority for wearable sensor based activity recognition. In addition, smartphones are less intrusive to subjects. For example, Dernbach et al. demonstrated the feasibility of using inertial sensor data that were collected from an android-base smart phone to recognize simple activities (e.g., biking, climbing, driving, lying, sitting, walking, running, and standing) and complex activities (e.g., cleaning, cooking, medication, sweeping, washing, and watering) [7]. However, wearable sensor based methods require an individual to wear or carry one or more devices all the time, so it is difficult for them to be widely applied in residences. That is, they are essentially intrusive methods and may bring inconvenience to individuals when performing ADLs.

In contrast, environment interactive sensors with the inherent non-intrusiveness have been proved applicable to the smart home setting when privacy and user acceptance are concerned [23]. For example, Tapia et al. proposed to build an activity recognition system that were installed with a set of simple state-change sensors, and deployed the system in two houses equipped with seventy-seven and eighty-four sensors, respectively. They collected data for a period of fourteen days and experimentally showed the feasibility of activity recognition with their system [23]. Similarly, van Kasteren et al. conducted a research to recognize seven different activities in a home setting via fourteen binary sensors and they obtained an accuracy of 79.4% [26].

On the other hand, to better capture the relations between the streaming sensor data and activity labels, researchers have proposed a wealth of models that range from discriminative models (e.g., conditional random field [6], support vector machine [9], and decision tree) to generative models (e.g., Naïve Bayes [23], hidden Markov model, and Bayesian networks

[31]). One common characteristic of these models is that they have a shallow structure and probably fail to capture non-linear relations hidden among features [2]. Moreover, researchers tend to extract an over-complete and discriminant set of features from the raw sensor data for analyzing complex human activities, while traditional methods rely heavily on domain knowledge to extract features and few studies consider learning latent features [20]. In addition, the selection of features and the training of classifiers are taken as two separate stages and not jointly optimized for these shallow models.

In contrast to the models with a shallow structure, deep learning models can automatically learn high-level features from the original low-level features rather than rely on the hand-crafted features. Consequently, deep learning techniques have been applied in many fields (e.g., speech recognition, image classification, and cancer diagnosis) and achieved great success. Moreover, there are studies that explore deep learn techniques to analyze sensor signals for human activity recognition. For example, Plötz et al. are among the first researchers that propose to develop an activity recognition framework that integrates principal component analysis and deep belief networks (DBN) for feature learning in ubiquitous computing [20]. Their experimental results on four publicly available datasets show the benefits of feature learning in activity recognition. Particularly, the basic unit of DBN provides a way to non-linearly transform the original input for feature learning. Also, Wang proposed to apply autoencoders to recognize human activities with accelerometer, gyroscope, and magnetometer [29], and achieved a recognition rate of 99.3%. To better capture the temporal dynamics of human activities, Ordóñez and Roggen proposed to construct convolutional LSTM recurrent neural networks for activity recognition [17], which is suitable for both homogeneous and heterogeneous wearable sensor data analyses. Because time series sensor data have the local dependency characteristics of sensor readings, Ronao and Cho built a deep convolutional neural network (CNN) to analyze smartphone sensor data for efficient and effective activity recognition [21]. They then conducted experiments on the benchmarked dataset and achieved an accuracy of 94.8%. Also, Yang et al. applied deep convolutional neural networks to analyze multichannel time series data for human activity recognition and obtained good performance [32]. However, although researchers have conducted considerable work on activity recognition with deep learning techniques, few studies, to the best of our knowledge, focus on the smart home setting that is equipped with simple state-change sensors. In contrast to the case of wearable sensors with a fixed sampling rate, these simple sensors are triggered by specific interactions and their firings have binary values. Besides, most studies consider one original feature representation of the raw sensor data and fail to explore the case of using at least two different schemes to describe sensor firings. Consequently, this study aims to use deep learning techniques for latent feature learning from simple state-change sensors and evaluate the role of different original feature representations in deep learning based activity recognizers.

3 Proposed model for activity recognition

3.1 Proposed activity recognition system

There are various classification models available for use to develop an activity recognizer. According to the availability of data labels, we broadly group them into three categories: supervised learning models working with labeled data, unsupervised learning models without data labels, and semi-supervised learning that can utilize both labeled and unlabeled data [4].

With the aim of pursuing a high accuracy, supervised learning with an explicit training phase is commonly used for human activity recognition and it typically consists of three steps. First, a stream of sensor data is divided into segments where the sliding window technique is used. Specifically, a window with a fixed/varied time-length or a fixed/varied number of sensor events is shifted along the stream with (non-) overlapping between adjacent segments. Second, we extract features from the segments to represent raw sensor data and then train a classifier using these obtained feature vectors. Third, we use the trained classifier to associate a stream of sensor data with a predefined activity. Figure 1 illustrates a schematic diagram of the proposed activity recognition system that mainly consists of the training phase and the test phase. In this study, we focus on a smart home setting equipped with various environmental sensors to infer the ongoing activity performed by an individual.

In the training phase (see steps 0–4 of Fig. 1), we first collect raw sensor data and annotate them with corresponding activity labels. We then divide the training set into segments and extract features from each segment to form a feature vector in the way as explained in *subsection B*. Because the quality of a feature representation largely determines the performance of an activity recognizer, we conduct latent feature learning with deep and hierarchy stacked autoencoders to explore non-linear relations inherent in low-level features. We detail corresponding procedures in *subsections C-E*. After the above steps, we get an optimized activity recognizer and use it for activity recognition (see steps a-d of Fig. 1). Specifically, we extract original features from the streaming sensor data, learn the latent high-level representation, and apply the trained recognizer to predict corresponding activity label.

3.2 Feature extraction

For sensor-based activity recognition, because standard classification models are not suitable for handling time series data, we are required to divide the streaming sensor data into segments

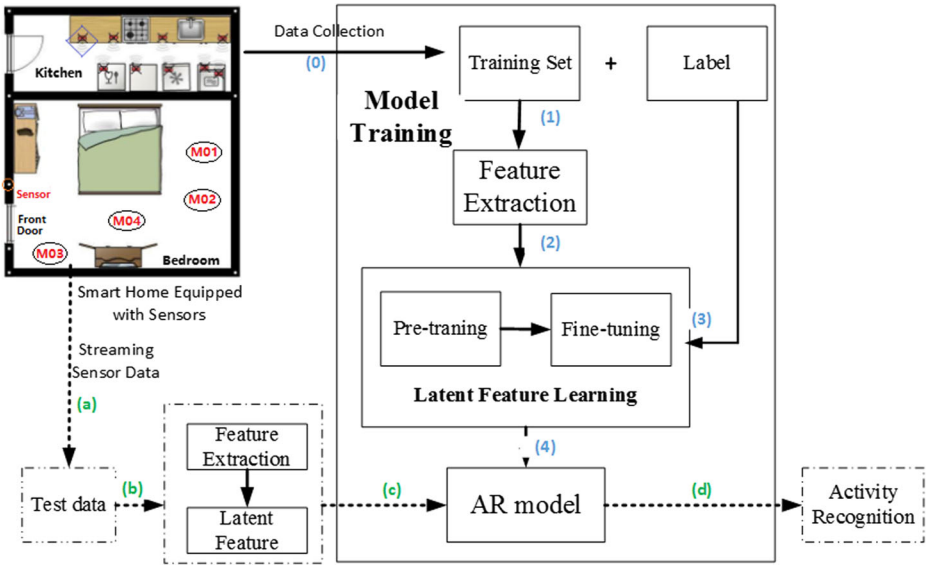

Fig. 1 Illustration of the proposed system for activity recognition. Steps (**0**)–(**4**) represent the model training phase, and steps (**a**)–(**d**) correspond to the activity recognition phase

before feature extraction. Sliding window technique is one of the widely used strategies. Generally, there are three common schemes to divide streaming data into segments: explicit segmentation, time-based segmentation, and sensor event-based segmentation [13]. Figure 2 is an abstract illustration of the three schemes. The sensor event sequence corresponds to sensor firings associated with three activities (activity 1, activity 2, and activity 3). Each symbol in the sensor sequence indicates a firing and is associated with a specific sensor. Herein, Fig. 2 indicates that there are six different sensors deployed in an indoor environment. For explicit segmentation, it segments streaming sensor events into chunks and obtains segments related to specific activities. An activity recognizer works well on the pre-segmented sequences of activity (e.g., P_1, P_2, and P_3), but it is difficult to determine an appropriate chunk size that can better capture the sensor firings of an activity. Consequently, this lowers the performance of an activity recognizer and makes it unsuitable for the online scenario. In contrast to explicit segmentation, time-based segmentation divides the entire sequence of streaming sensor events into time intervals of equal size, such as T_1, T_2, and T_9 in Fig. 2. This scheme provides a simpler way to segment time series data and has been widely used by researchers. For sensor event-based segmentation, it divides the sequence of sensor firings into segments that have equal number of sensor events. Obviously, the segments may have different durations because of the varying sampling rates of the deployed sensors. For example, segment S_3 has a larger duration than that of segments S_1 and S_2. Both time-based segmentation and sensor event-based segmentation require users to optimize the window size. A window of small size probably contains little information relevant to an activity, while a window of large size may include sensor events associated with multiple activities.

In addition, according to whether overlapping parts exist between two consecutive segments, sliding window technique has two classical schemes: sliding window with overlapping and sliding window without overlapping. Furthermore, according to whether the time-length or the number of sensor events in a window is fixed, we group sliding window techniques into fixed length category and varying length category [16]. For the former case, all segments have equal time length or contain the same number of sensor events. However, for the varying length case, some segments have different length of time or different number of sensor events. In this study, we use the time-based sliding window technique without overlapping between adjacent segments to analyze time series sensor data. After segmenting streaming sensor data,

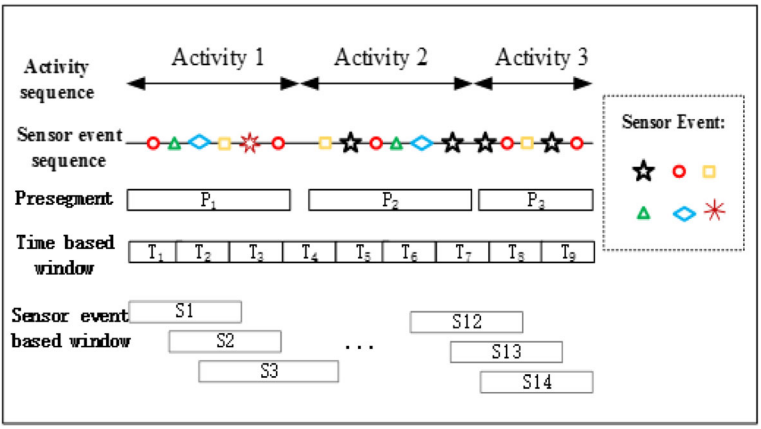

Fig. 2 Illustration of different schemes for segmenting sensor data

we extract features from each segment and use these features to form a N-dimensional feature vector $x_t = \left(x_t^1, x_t^2, \ldots, x_t^N\right)$, where N is total number of sensors installed in a smart home and each dimension of x_t corresponds to a physical sensor. In particular, we use two different original feature representations to represent the raw sensor data: binary representation and numerical representation. The numerical representation records the number of firings of a sensor during a specific time interval, while binary representation records whether a sensor fired at least once during a time slice. That is, binary representation takes the value of one if corresponding sensor fired; otherwise zero. For example, given a smart home equipped with four state-change sensors, the resident performed an activity "make tea" at time t, which triggered the first sensor two times, the second sensor once and the fourth sensor three times. We have $x_t = (1, 1, 0, 1)$ for binary representation and $x_t = (2, 1, 0, 3)$ for numerical representation. After obtaining the original feature representation, we can exploit the deep learning technique to learn the latent features, as discussed in the following two sections.

3.3 Autoencoder

An autoencoder is essentially an artificial neural network that consists of three layers: input layer, hidden layer, and output layer (see Fig. 3), and it aims to learn a latent representation $h(x)$ of the input vector x with an extra constraint that the target values of the output layer are equal to or approximate to input values of the input layer. Suppose that N and K denote the number of input units and hidden units, respectively, an autoencoder transforms an N-dimensional vector $x = (x_1, x_2, \ldots, x_N)$ to a latent representation $h(x) = (h_1, h_2, \ldots, h_k) \in \mathrm{R}^{K \times 1}$ through a deterministic mapping (Eq. 1), also called encoding,

$$h(x) = f\left(W^{(1)}x + b^{(1)}\right) \tag{1}$$

, where $W^{(1)} \in \mathrm{R}^{K \times N}$ is a matrix that stores the weights from input units to hidden units, $b^{(1)}$ represents the biases of hidden units, and $f(\cdot)$ is the activation function of each unit. Sigmoid function is one of the most commonly used non-linear activation functions and shown mathematically as Eq. (2).

$$f(x) = 1/(1 + \exp(-x)) \tag{2}$$

We then reconstruct x from its latent representation $h(x)$ using Eq. 3 under the constraint of trying to minimize the difference between x and x'.

$$x' = f\left(W^{(2)}x + b^{(2)}\right) \tag{3}$$

, where $x' = \left(x_1', x_2', \ldots, x_N'\right)$, $W^{(2)} \in \mathrm{R}^{N \times K}$ stores the weights from hidden units to output units, and $b^{(2)}$ represents the biases of output units. This enables us to obtain a new feature representation $h(x)$ of the original feature x. In particular, the number of units in the hidden layer can be larger or less than the input dimension.

To obtain a robust feature representation, on the basis of autoencoder, Vincent et al. proposed a denoising autoencoder model that reconstructs the original input from their corrupted version with a local denosing criterion [28]. In practice, the corrupted input can be generated by adding random noises to the original input or randomly setting values of the original input to be zero. In this study, we use the denoising autoencoder as building blocks of the proposed activity recognizer.

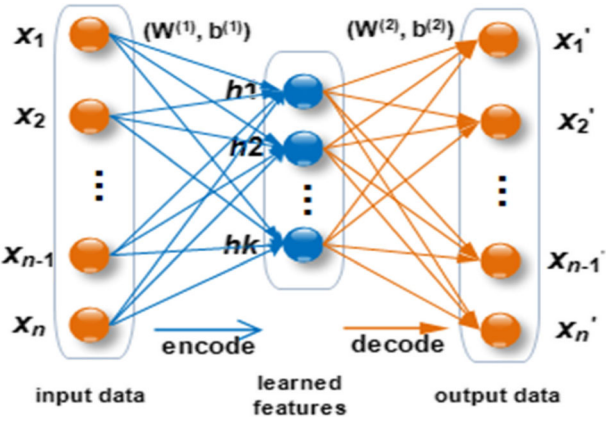

Fig. 3 Architecture of an autoencoder

3.4 Stacked autoencoder

Recent advances in deep learning show that a deep and hierarchical architecture helps obtain complex and non-linear latent relations in comparison to the shallow structures that have no or only one hidden layer. A stacked autoencoder (SAE) is such a hierarchical model that consists of autoencoders [10, 28]. Each layer of a SAE is fully connected to its adjacent layer, and there is no connection between units in the same layer. Similar to the autoencoder, each hidden layer of SAE is a high-level representation of the original input and the objective function of SAE is to reconstruct the input at the output layer. The number of units in a hidden layer can be equal to, larger or less than the input dimension. This enables us to sufficiently and flexibly explore different high-level features.

Due to the deep structure, conventional gradient-based optimization methods (e.g., SGD and L-BFGS) suffer from the gradient diffusion in training a stacked autoencoder and are easily trapped into a poor local optimum on a network with randomly initialized weights and biases. To alleviate this problem and improve the convergence rate, Hinton et al. proposed a greedy layer-wise learning process to train a deep belief network and experimentally showed its good performance [11]. Such a strategy trains each sub-network separately rather than trains the whole network simultaneously, and it directs the output of one sub-network as the input of its next sub-network. Specifically, we use the training data as input of an autoencoder to learn the first hidden layer, then use the first hidden layer as input to learn the second layer, and so on. Generally, suppose there is a stacked autoencoder with n layers and the first layer is the original training data. For the m-th autoencoder $1 \le m \le n$, $W^{(m)}$ stores the weights between input units and hidden units and $b^{(m)}$ is the biases of hidden layer. The greedy layer-wise scheme performs the following two steps iteratively.

$$a^{(m)} = f\left(z^{(m)}\right) \tag{4}$$

$$z^{(m+1)} = W^{(m)} a^{(m)} + b^{(m)} \tag{5}$$

, where $z^{(m)}$ is the input of the m-th layer, $a^{(m)}$ is the activation of the m-th layer, and $a^{(1)} = x$ when $m = 1$. Obviously, $a^{(n)}$ is the inner-most feature representation. The above process is called *pre-training*, and it works without labeled data.

3.5 Fine-tuning activity recognition model

With the learned features of a SAE, we can combine it with a set of labeled data to build a classifier. Accordingly, we stack another output layer, called classifier layer, on top of a stacked autoencoder to train a classifier. That is, the feature vector encoded in the last hidden layer is the input of a learning algorithm in the classifier layer, and we can use different classification models. Figure 4 presents the overall architecture of an activity recognition model with a softmax classifier. The stacked autoencoder is used for latent learning, the last layer is the classifier layer, and the number of units of it is equal to the number of activities of interest. For Fig. 4, the probability output $p(y = c|x)$ determines the label of a sample x, where c indicates the c-th activity that we have predefined. x_1, x_2, ..., x_{n-1} and x_n are the input, each hidden layer is a high-level representation of the original data, and the last hidden layer is retained as the input of the classifier layer.

With the availability of labeled data, we can optimize the activity recognition model in a supervised manner. Specifically, we first initialize weights and biases of a network with values obtained by the *pre-training*, and then use the backpropagation algorithm with gradient descent to further optimize model parameters. Previous researches show that such a strategy helps escape from the poor local optimum and improve training time performance [3]. This procedure benefits from labeled data and is called *fine-tuning*.

To a further step, to determine the optimal values of hyperparameters (e.g., the percentage of masking noise and learning rate) and the network layout (e.g., the number of hidden layers and the number of units in each hidden layer), besides fine-tuning, we employ a grid-based search strategy and cross-validation to choose the best network structure as the activity recognizer. The pseudo code of the proposed activity recognition algorithm is presented in Table 1. Y_l is the label matrix for labeled data. We let $Y_l(i,k)$ be 1 if $X_l^{(i)}$ is labeled as class k, and 0 otherwise. h denotes the number of autoencoders of SAE and u_i $(1 \le i \le h)$ is the number of hidden units of the i^{th} autoencoder.

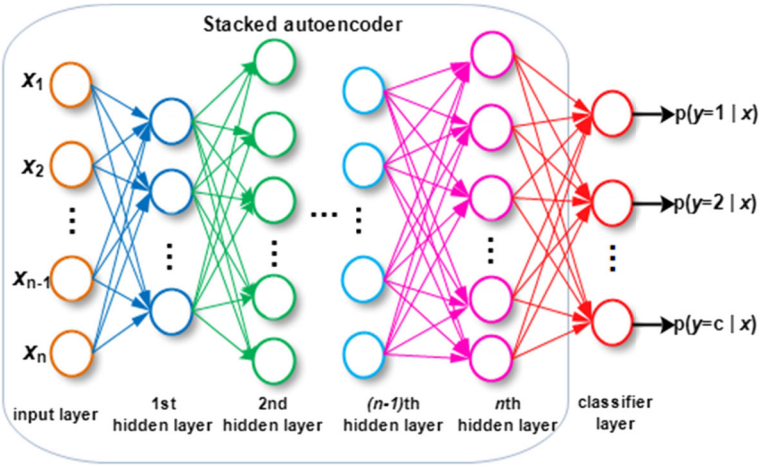

Fig. 4 Illustration to an activity recognition model with a stacked autoencoder and a softmax classifier

Table 1 Algorithm of the proposed activity recognizer

Input	Labeled data X_l, unlabeled data X_u, Y_l is the label matrix of X_l, the number of autoencoders h, vector $u = [u_1, u_2, ..., u_h]$ containing the number of hidden units for corresponding autoencoder, activation function f
Output	Activity recognizer SDA_CLS
1	// **Initialization**
	1.1 initialize network structure of a stacked denoising autoencoder SDA with given h and u
	1.2 initialize constants and parameters of SDA randomly
2	// **Pre-training with unlabeled data**
	2.1 train each denoising autoencoder of SDA layer-wisely as follows:
	a) train the first autoencoder with X_l and X_u as input
	b) train the i^{th} $(2 \le i \le h)$ autoencoder with the $(i\text{-}1)^{th}$ autoencoder as input
3	// **Initialization**
	3.1 stack a classifier layer on top of SDA, and call the network structure SDA_CLS
	3.2 initialize weights of the classifier layer randomly
4	// **Fine-tuning with labeled data**
	4.1 take X_l and Y_l as input of SDA_CLS and predict corresponding output
	4.2 iteratively tune network weights with backpropagation algorithm by minimizing the given cost function
5	**return** SDA_CLS

4 Experimental results and analysis

4.1 Experimental datasets

To evaluate the performance of deep learning based activity recognition model and evaluate the role of different original feature representations, we conduct experiments on three publicly available datasets that were collected from three smart homes equipped with simple sensors. Each of the smart homes housed one resident performing ADLs in it. For the first smart home (D1), there are three rooms equipped with 14 sensors in total. Sensor data was collected over a period of twenty-five days and ten activities were observed, resulting in 1229 sensor events and 292 activity instances. For the second smart home (D2), 13 activities were observed during a period of fourteen days in an apartment that was installed with twenty-three sensors. There are 200 activity instances and 19,075 sensor events. The third smart home (D3) was monitored for nineteen days, and 344 activity instances and 22,700 sensor events were collected. A summary of the experimental datasets is given in Table 2 (please refer to [25] for details). Notably, all sensors involved, such as motion detector sensors, contact switch sensors, pressure mats, mercury contacts, and float sensors, are simple state-change sensors.

4.2 Experimental setup and results

In this study, time-based segmentation is used and the streaming sensor data is first divided into segments by shifting a fixed length, non-overlapping sliding window of sixty seconds as suggested in [12]. To evaluate the quality of the constructed AR model, we performed leave one day out cross validation, where one full day of sensor data is used as a test set and sensor data from the remaining days are used for classifier training. We repeat the above process the number of days times and report the performance of a classifier in terms of time-slice accuracy and class accuracy [12].

$$timeslice\_accuracy = \frac{\sum_{n=1}^{N} I(inferred(n) == true(n))}{N} \tag{6}$$

Table 2 Experimental dataset description

Dataset setting	D1 apartment	D2 apartment	D3 house
#resident	1	1	1
resident age	26	28	57
#rooms	3	2	6
#days monitored	25	14	19
#sensors	14	23	21
#activities	10	13	16
#sensor events	1229	19,075	22,700
#activity instances	292	200	344

$$class\_accuracy = \frac{1}{C}\sum_{c=1}^{C}\left\{\frac{\sum_{n=1}^{N_c} I(inferred(n) == true(n))}{N_c}\right\} \tag{7}$$

, where $I(a == b)$ is a indicator function returning 1 if a equals b and 0 otherwise, N is the total number of segments of a test set, N_c denotes the number of segments belonging to class c, C is the number of activities, $inferred(n)$ is the inferred label of segment n, and $true(n)$ is the true label of segment n. In this study, rather than explore a large number of autoencoders, we use two stacked autoencoders with different number of hidden layers to learn latent features in the proposed AR model: one-layer denoising autoencoder (DAE) and two-layer stacked denoising autoencoder (SDAE). Besides, we set the number of units in the hidden layer ranging from five to one hundred with a step size of five, and set the percentage of masking noise being 0.5. In addition, we compare DAE and SDAE with other five commonly used baselines, including Naïve bayes (NB), hidden markov model (HMM), hidden semi-markov model (HSMM), 1-nearest-neighbor (KNN), and support vector machine with linear kernel (SVM). All these five predictors have a shallow structure and they directly use the original binary representation or numerical representation as input. For KNN, we use one nearest neighbor (1NN) to determine the label of a test sample.

For each dataset, we studied two different original feature representations and reported the average time-slice accuracy and class accuracy of the leave one day out cross validation. Experimental results on the three datasets are presented in Tables 3, 4, and 5, repectively. The first column "Feature" indicates the original feature representation with "Binary" ("Numerical") corresponding to binary representation (numerical representation). From Table 3, we observe that SDAE outperforms other six methods in terms of time-slice accuracy and class accuracy metrics for both binary representation and numerical representation. Specifically, SADE obtains a time-slice accuracy of 85.32% and a class accuracy of 49.91% in comparison to the 59.11% time-slice accuracy and 48.46% class accuracy of the commonly used HMM in the case of binary representation. For numerical representation, SDAE achieves

Table 3 Experimental results on D1

Feature	Metric (%)	NB	HMM	HSMM	1NN	SVM	DAE	SDAE
Binary	Time-slice	77.14	59.11	59.46	33.10	83.88	85.26	85.32
	Class	42.62	45.48	48.46	32.43	48.14	48.02	49.91
Numerical	Time-slice	77.03	59.73	60.10	33.30	83.95	85.51	85.52
	Class	38.43	43.35	44.15	33.06	48.18	52.82	53.42

Table 4 Experimental results on D2

Feature	Metric (%)	NB	HMM	HSMM	1NN	SVM	DAE	SDAE
Binary	Time-slice	80.35	63.23	63.83	55.73	82.60	84.37	84.16
	Class	32.47	44.66	44.60	30.47	41.76	36.16	39.92
Numerical	Time-slice	80.50	66.79	69.42	59.03	81.82	85.06	85.61
	Class	24.83	28.79	31.22	39.46	44.51	40.39	43.30

a time-slice accuracy of 85.52% and a class accuracy of 53.42%, while HMM obtains a 59.73% time-slice accuracy and a 43.3% class accuracy. We can see that HSMM achieves similar performance to HMM. For NB that is built on the basis of conditional independence among features, it performs poorly whichever feature representation is used. This indicates that there exist latent relations among the original features. Also, the instance-based method KNN fails to achieve good results. From Table 4, we observe that deep learning based methods outperform the competitors in terms of time-slice accuracy and that their differences in class accuracy is quite small. For example, SDAE achieves a class accuracy of 43.30%, which is 1.21% less than 44.51% of SVM for numerical representation. In term of time-slice accuracy, SDAE has an accuracy of 85.61% in comparison to 81.82% of SVM. Similar conclusions can be drawn from Table 5. Overall, the above experimental results demonstrate the effectiveness of the proposed deep learning based model in human activity recognition. Particularly, it should be noted that in this study, we only explore a deep learning architecture with at most two layers and small number of units in each layer. Therefore, we do not fully explore the power of latent feature learning.

To present a better comparison of these methods and investigate their behaviors in two different original feature representations, Fig. 5a shows the time-slice accuracy averaged over all the datasets for both binary representation and numerical representation, and Fig. 5b presents corresponding results of class accuracy. We observe the following results. (1) Deep learning based methods (DAE and SDAE) outperform other five competing methods in terms of time-slice accuracy. For class accuracy, deep learning based methods outperform NB, HMM, HSMM and 1NN using numerical representation and obtain similar accuracies to other methods with binary representation. (2) SDAE with two hidden layers obtains a higher accuracy than DAE with one hidden layer. This indicates that a deeper model probably contributes to the improvement of an AR system. (3) In contrast to the model with a shallow structure, deep learning based methods are robust across different original feature representations. For example, HMM obtains a class accuracy of 35.8% with binary representation, which is decreased by 8.0% compared to the case of numerical representation. For HSMM, it achieves a class accuracy of 29.8% for numerical representation in comparison to the 37.8% of binary representation. This indicates that deep learning based model generalizes better

Table 5 Experimental results on D3

Feature	Metric (%)	NB	HMM	HSMM	1NN	SVM	DAE	SDAE
Binary	Time-slice	46.47	26.48	31.17	27.73	44.56	44.45	50.04
	Class	16.84	17.22	20.35	19.81	21.33	19.24	21.14
Numerical	Time-slice	41.44	27.37	32.31	30.82	42.70	46.84	54.82
	Class	11.17	11.21	14.06	24.98	25.45	23.96	22.08

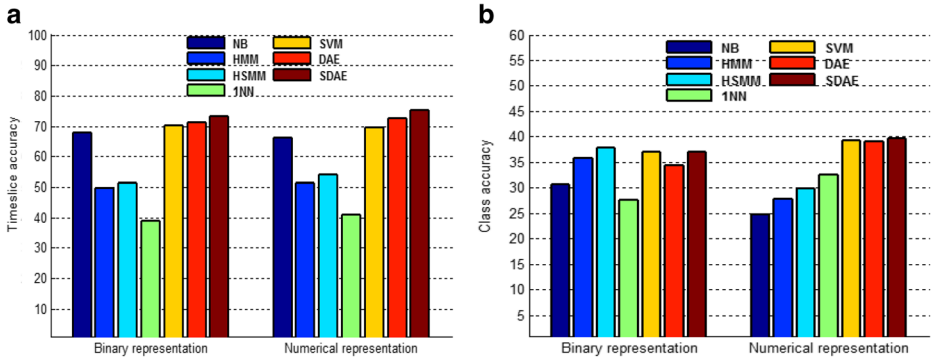

Fig. 5 **a** Comparison in time-accuracy of the seven methods **b** Comparison in class accuracy of the seven methods

across different original feature representations and can potentially relieve users of the reliance on domain knowledge to hand-craft features.

Furthermore, we preliminarily investigate another commonly used deep learning technique named Deep Belief Network (DBN) [11] and compare it with SDAE. Similar to SDAE, DBN is also a stack model and a Restricted Boltzman Mahince (RBM) is its building block. An RBM is a bipartite graphical model with full connections between two layers. We train a DBN layer-wisely, and fine-tuning is performed via supervised gradient descent. For a fair comparison, we use a DBN with two hidden layers and conduct initial experiments on the three above mentioned datasets. Table 6 presents experimental results of SDAE and DBN. We observe that SDAE obtains higher accuracy than DBN, especially for the case of numerical representation. In particular, we observe that SDAE generalizes better across different original feature representations in comparison with DBN. For example, DBN obtains a time-slice accuracy of 82.91% for binary representation on D1, while the time-slice accuracy decreases to 54.28% for numerical representation.

5 Conclusion and future work

Pervasive computing technology has been gradually applied in smart homes for human-centric applications due to its non-intrusiveness, low cost, and easy deployment. For human activity recognition, researchers have conducted considerable work in exploring various sensing units and developing different models. However, few studies explore to automatically learn high-

Table 6 Experimental results of SDAE and DBN

Dataset	model	binary representation		numerical representation	
		time-slice (%)	class (%)	time-slice (%)	class (%)
D1	SDAE	85.32	49.91	85.52	53.42
	DBN	82.91	32.70	54.28	20.84
D2	SDAE	84.16	39.92	85.61	43.30
	DBN	83.14	29.45	60.23	16.58
D3	SDAE	50.04	21.14	54.82	22.08
	DBN	42.48	16.37	46.17	12.52

level features and to jointly optimize feature extraction and classifier training. In this study, we construct a deep and hierarchical autoencoder model to learn latent features from the raw sensor data for human activity recognition. Specifically, we first illustrated three different segmentation schemes for dividing streaming sensor data. Then, we detailed how to construct a deep learning based model for feature learning and classifier training. To demonstrate the effectiveness of SDAE and DAE, we conducted experiments on three publicly available smart home datasets and compared it with other five traditional methods in terms of time-slice accuracy and class accuracy metrics. In addition, we explore two different original feature representations of the raw sensor data, and experimentally evaluate their roles in deep learning models and shallow models. Experimental results show that deep learning based method outperforms its competitors and generalizes better across different feature representations.

For the future work, we plan to do in the following lines. First, in this study, we only explore the stacked autoencoder model for human activity recognition and preliminarily compare the effectiveness of DBN and SDAE. There are numerous deep learning models available such as convolutional neural networks and recurrent neural networks, and we plan to investigate them in a smart home setting. Second, this study only focuses on single resident activity recognition, so multi-resident activity recognition remains another research topic.

Acknowledgements This work was supported partially by the Natural Science Foundation of China (No. 61472057), the Fundamental Research Funds for the Central Universities (No. JZ2016HGBH1053), and the China Postdoctoral Science Foundation (No. 2016 M592046).

Compliance with ethical standards

Conflict of interest The authors claim no conflict of interest.

References

1. Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. In: International Conference on Pervasive Computing, pp 1–17
2. Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2(1):1–127. doi:10.1561/2200000006
3. Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, pp 153–160
4. Bhattacharya S, Nurmi P, Hammerla N, Plötz T (2014) Using unlabeled data in a sparse-coding framework for human activity recognition. Pervasive Mob Comput 15:242–262. doi:10.1016/j.pmcj.2014.05.006
5. Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C 42(6):790–808. doi:10.1109/TSMCC.2012.2198883
6. Cook DJ (2012) Learning setting-generalized activity models for smart spaces. IEEE Intell Syst 27(1):32–38. doi:10.1109/MIS.2010.112
7. Dernbach S, Das B, Krishnan NC, Thomas BL, Cook DJ (2012) Simple and complex activity recognition through smart phones. In: 2012 8th International Conference on Intelligent Environments, pp 214–221
8. Figo D, Diniz PC, Ferreira DR, Cardoso JM (2010) Preprocessing techniques for context recognition from accelerometer data. Pers Ubiquit Comput 14(7):645–662. doi:10.1007/s00779-010-0293-9
9. Fleury A, Vacher M, Noury N (2010) SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. IEEE Trans Inf Tech Biomed 14(2):274–283. doi:10.1109/TITB.2009.2037317
10. Hinton G, Salakhutdinov R (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507. doi:10.1126/science.1127647

11. Hinton G, Osindero S, Teh Y (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7): 1527–1554. doi:10.1162/neco.2006.18.7.1527
12. Kim SC, Jeong YS, Park SO (2013) RFID-based indoor location tracking to ensure the safety of the elderly in smart home environments. Pers Ubiquit Comput 17(8):1699–1707. doi:10.1007/s00779-012-0604-4
13. Krishnan NC, Cook DJ (2014) Activity recognition on streaming sensor data. Pervasive Mob Comput 10: 138–154. doi:10.1016/j.pmcj.2012.07.003
14. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. ACM SigKDD Explor News 12(2):74–82. doi:10.1145/1964897.1964918
15. Minor B, Doppa JR, Cook DJ (2015) Data-driven activity prediction: algorithms, evaluation methodology, and applications. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 805–814
16. Okeyo G, Chen L, Wang H, Sterritt R (2014) Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. Pervasive Mob Comput 10:155–172. doi:10.1016/j.pmcj.2012.11.004
17. Ordóñez FJ, Roggen D (2016) Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115. doi:10.3390/s16010115
18. Ordóñez F, de Toledo P, Sanchis A (2015) Sensor-based bayesian detection of anomalous living patterns in a home setting. Pers Ubiquit Comput 19(2):259–270. doi:10.1007/s00779-014-0820-1
19. Philipose M, Fishkin KP, Perkowitz M, Patterson DJ, Fox D, Kautz H, Hähnel D (2004) Inferring activities from interactions with objects. IEEE Pervasive Comput 3(4):50–57. doi:10.1109/MPRV.2004.7
20. Plötz T, Hammerla NY, Olivier P (2011) Feature learning for activity recognition in ubiquitous computing. In: Proceedings International Joint Conference on Artificial Intelligence, pp 1729–1734
21. Ronao CA, Cho SB (2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244. doi:10.1016/j.eswa.2016.04.032
22. Suryadevara NK, Mukhopadhyay SC (2014) Determining wellness through an ambient assisted living environment. IEEE Intell Syst 29(3):30–37. doi:10.1109/MIS.2014.16
23. Tapia EM, Intille SS, Larson K (2004) Activity recognition in the home using simple and ubiquitous sensors. In: International Conference on Pervasive Computing, pp 158–175
24. Tapia E, Intille S, Haskell W, Larson K, Wright J, King A, Friedman R (2007) Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In: 11th IEEE International Symposium on Wearable Computers, pp 37–40
25. van Kasteren TLM (2011) Activity recognition for health monitoring elderly using temporal probabilistic models. Dissertation, University of Amsterdam
26. Van Kasteren T, Noulas A, Englebienne G, Kröse B (2008) Accurate activity recognition in a home setting. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp 1–9
27. van Kasteren TLM, Englebienne G, Kröse BJ (2010) An activity monitoring system for elderly care using generative and discriminative models. Pers Ubiquit Comput 14(6):489–498. doi:10.1007/s00779-009-0277-9
28. Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp 1096–1103
29. Wang L (2016) Recognition of human activities using continuous autoencoders with wearable sensors. Sensors 16(2):189. doi:10.3390/s16020189
30. Wang A, Chen G, Yang J, Zhao S, Chang CY (2016) A comparative study on human activity recognition using inertial sensors in a smartphone. IEEE Sensors J 16(11):4566–4578. doi:10.1109/JSEN.2016.2545708
31. Wilson DH, Atkeson C (2005) Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In: International Conference on Pervasive Computing, pp 62–79
32. Yang J, Nguyen M, San P, Li X, Krishnaswamy S (2015) Deep convolutional neural networks on multichannel time series for human activity recognition. In: Proceedings International Joint Conference on Artificial Intelligence, pp 3995–4001

Guilin Chen is currently a Professor with the School of Computer and Information Engineering, Chuzhou University, Anhui, China. He received the B.S. degree from Anhui Normal University, China, in 1985, and the M.S. degree from the Hefei University of Technology, in 2007. His current research interests include cloud computing, wireless networks, healthcare, and Internet of Things.

Aiguo Wang is a post doctor with School of Computer and Information, Hefei University of Technology, China. He received his Bachelor's degree and Ph.D. degree at Hefei University of Technology in 2010 and 2015, respectively. His research interests include data mining, bioinformatics, activity recognition, and ambient assisted living.

Shenghui Zhao is currently a Professor with the School of Computer and Information Engineering, Chuzhou University, China. She received the M.S. degree from the Hefei University of Technology, China, in 2003, and the Ph.D. degree from Southeast University, China, in 2013. Her current research interests include trusted computing, wireless networks, healthcare, and Internet of Things.

Li Liu is an associate professor of School of Software Engineering at Chongqing University. He is also serving as a Senior Research Fellow of School of Computing at the National University of Singapore. Li received his Ph.D. in Computer Science from the University Paris-sud 11. He had served as an associate professor at Lanzhou University in China. His research interests are in mobile and ubiquitous computing, data analysis, and their applications on health and behavior. He aims to contribute in interdisciplinary research of computer science and human related disciplines. Li has published widely in conferences and journals with more than 50 peer-reviewed publications. Li has been the Principal Investigator of several funded projects from government and industry.

Chih-Yung Chang is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan. He received the Ph.D. degree in computer science and information engineering from National Central University, Zhongli, Taiwan, in 1995. His current research interests include Internet of Things, wireless sensor networks, ad hoc wireless networks, and long-term evolution broadband technologies. He served as an Associate Guest Editor of several SCI-indexed journals, including the *International Journal of* Ad Hoc *and Ubiquitous Computing* from 2011 to 2014, the *International Journal of Distributed Sensor Networks* from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* in 2010, the *Journal of Information Science and Engineering* in 2008, and the *Journal of Internet Technology* from 2004 to 2008.