Human Activity Recognition in a Smart Home Environment with Stacked Denoising Autoencoders

Aiguo Wang[1,2], Guilin Chen[1(✉)], Cuijuan Shang[1], Miaofei Zhang[1], and Li Liu[3]

[1] School of Computer and Information Engineering, Chuzhou University,
Chuzhou 239000, China
{glchen,shangcuijuan,zhangmiaofei}@chzu.edu.cn,
wangaiguo2546@163.com

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

[3] School of Software Engineering, Chongqing University, Chongqing 400044, China
dcsliuli@cqu.edu.cn

Abstract. Activity recognition is an important step towards automatically measuring the functional health of individuals in smart home settings. Since the inherent nature of human activities is characterized by a high degree of complexity and uncertainty, it poses a great challenge to build a robust activity recognition model. This study aims to exploit deep learning techniques to learn high-level features from the binary sensor data under the assumption that there exist discriminant latent patterns inherent in the low-level features. Specifically, we first adopt a stacked autoencoder to extract high-level features, and then integrate feature extraction and classifier training into a unified framework to obtain a jointly optimized activity recognizer. We use three benchmark datasets to evaluate our method, and investigate two different original sensor data representations. Experimental results show that the proposed method achieves better recognition rate and generalizes better across different original feature representations compared with other four competing methods.

Keywords: Activity recognition · Smart homes · Deep learning · Autoencoder · Shallow structure model

1 Introduction

The rapid development of machine learning and mobile computing technologies makes it possible for researchers to customize and provide pervasive and context-aware services to individuals living in smart homes [1]. On the other hand, due to the ever increasing aging population all over the world and the high expenditure of healthcare cost, the elderly healthcare raises us a serious social and fiscal problem. With the growing desire of subjects to remain independent in their own homes, ambient assisted living (AAL) systems that can perceive the states of an individual and corresponding context and act on physical surroundings using different types of sensors and automatically recognize human activities of daily living (ADLs) are in great needs [2, 3]. In such systems, accurately recognizing human activities such as cooking, eating, drinking, grooming and sleeping is an important step towards independent living, which can be achieved by

S. Song and Y. Tong (Eds.): WAIM 2016 Workshops, LNCS 9998, pp. 29–40, 2016.
DOI: 10.1007/978-3-319-47121-1_3

monitoring the function ability of the residents using various sensor technologies. Also, activity recognition can potentially facilitate a number of applications in a home setting such as fall detection, activity reminder, and welling evaluation [4, 5].

Activity recognition (AR) is a challenging and active research area [6], and different types of sensing technologies have been explored by researchers to improve the recognition rate and adapt to different application scenarios. Generally, they can be mainly grouped into three categories: vision-based (e.g. camera, video), wearable/carriable sensor-based (e.g. accelerometer, gyroscope), and environment interactive sensor-based methods (e.g. motion detector, pressure sensor, contact sensor) [7, 8]. Due to the inherent non-intrusiveness, flexibility, low cost, and easy deployment, environment sensor-based approaches are considered a promising way to assess individual physical and cognitive health when privacy and user acceptance issues are considered [1]. Approaches belonging to this category infer the ADLs performed by an individual by capturing the interactions between an individual and a specific object. For example, we can use a contact sensor to record whenever the medicine container is open or closed for the application of adherence to medication. In sensor-based activity recognition, the output of an AR system is a stream of sensor activations [7, 9]. We can then treat activity recognition as a time series analysis problem, and the aim is to identify a continuous portion of sensor data stream associated with one of the preselected known activities. The widely used approach to AR is to apply the supervised learning with an explicit training phase, which mainly consists of three stages [10, 11]. First, a stream of sensor data is divided into segments, in which a sliding window technique is often used. Specifically, a window with a fixed time length or fixed number of sensor events is shifted along the stream with (non-) overlapping between adjacent segments. The next step is to extract features from the segments and transform the raw signal data into feature vectors, followed by the classifier construction with these features. The last task, called recognition phase, is to use the trained classifier to associate a stream of sensor data with a predefined activity. From the view of pattern recognition and machine learning, appropriate feature representation of sensor data, suitable choice of classifier and its parameter settings are crucial factors that determine the performance of AR [12]. Although researchers have proposed a number of models to recognize ADLs, however, most of existing AR approaches usually rely on hand-crafted features such as mean, variance, correlation coefficients and entropy, and this may result in loss of information. Also, most classifiers used have been shown to have shallow structures, hence it is difficult for them to discover the latent non-linear relations inherent in features [13]. Furthermore, in most studies, feature extraction and classifier training are treated as two separate steps, so they are not jointly optimized. Consequently, without the guidance of classification performance, the best way to design and choose feature descriptors is not clear, and we may fail to obtain satisfactory accuracy without the exploration of feature extraction.

In recent years, deep learning techniques have gained great popularity and been successfully applied in various fields such as speech recognition and face recognition due to its representational power. These techniques enable the automatic extraction of features from the original low-level features without any specific domain knowledge but with a general-purpose learning procedure. In this study, to improve the activity recognition performance, we propose to exploit deep learning techniques to discover the latent useful information inherent in the original features, and integrate feature learning and classifier training into an architecture to jointly optimize them. Specifically, we use a denoising autoencoder to learn the underlying feature representation from unlabeled data, and the obtained features are then used as the inputs of a top classifier. This enables us to unify feature learning and classifier training in a single pipeline and further to fine-tune the model parameters using labeled data in order to obtain a robust model.

The rest of this paper is structured as follows. Section 2 briefly reviews related work in activity recognition. We then illustrate the autoencoder model, the pre-training and fine-tuning scheme, and the proposed activity model in Sect. 3. In Sect. 4, experimental setup and results are presented. The last section concludes this study with a short summary and discussion.

2 Related Work

To improve the performance of activity recognition and enable its wide applications in real world scenarios, researchers have conducted considerable work in exploring various sensing technologies and designing a number of methods to model and recognize human activities [7]. It has been shown that different types of sensor modalities are effective for recognizing different activities. Vision-based approaches can provide 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 [14]. Moreover, vision-based approaches face technical challenges arising from light, distance from cameras, occlusion and low object recognition rate, which largely hinder their wide use. In the past few years, due to the rapid development of information technology, a variety of sensors are designed and used for human activity recognition due to their flexibility, low cost, and less intrusiveness [15]. These sensors can be categorized into wearable sensors and environment interactive sensors. In the former case, commonly used sensors that can be worn or carriable include accelerometer, gyroscope, GPS, and RFID-readers (used together with RFID tags). For example, Bao and Intille used five small biaxial accelerometers that were worn simultaneously on different parts of the body to recognize twenty activities. By collecting experimental data from twenty volunteers and extracting time-domain and frequency-domain features, they compared the recognition rate of three different classifiers and showed that the decision tree algorithm achieved the best performance with an accuracy of 84.0 % [16]. With the increasing processing and communication power of mobiles devices, most smartphones that are embedded with built-in GPS, accelerometers and gyroscopes are used for activity recognition due to the fact that they are less intrusive to subjects and that no additional equipment is required for data collection and procession [17, 18]. For example, Dernbach et al. demonstrated the possibility of using the inertial sensor data collected from android-based smart phones to recognize simple activities such as biking, climbing, driving, lying, sitting, walking, running and standing, as well as complex activities such as cleaning, cooking, medication, sweeping, washing and watering [19]. Besides these, RFID technology provides a solution to activity recognition as well, because they can capture the interaction between an individual and the objects. For example, Kim et al. built an indoor healthcare monitoring system to locate and track the elderly in real time by capturing the interaction between subjects (an individual wearing a RFID reader) and the tagged objects with RFID technology [20]. Philipose et al. applied RFID technology, data mining and a probabilistic engine for fine-grained activity recognition based on the interaction between objects and subjects [9].

Although wearable sensor based approaches can obtain satisfactory performance, it is difficult for them to be widely applied in residences because this kind of method requires users to wear or carry corresponding sensors all the time. Therefore, they are actually intrusive and may bring inconvenience to individuals when performing ADLs. In contrast, environment interactive sensors with inherent non-intrusive characteristics have proven applicable to the home setting when privacy and user acceptance are concerned [1]. For example, Tapia et al. built an activity recognition system installed with a set of simple state-change sensors, and then deployed their system in two houses equipped with seventy-seven and eighty-four sensors, respectively, and collected data for fourteen days to show its feasibility in AR [1]. van Kasteren et al. carried out a research to recognize seven different activities in a home setting via fourteen binary sensors and obtained an accuracy of 79.4 % [21]. In different studies, several models have been used to recognize activity such as Naïve Bayes [1], hidden markov model [2], support vector machine [22], Bayesian networks [23], and sparse coding [24]. One common feature of these models is that they all have shallow structures and may not capture the complex non-linear relations among features [13]. Also, to analyze the complex human activities, it is expected to extract over-complete and discriminant features from sensor data, and traditional methods rely on domain knowledge to extract features and few consider to learn features from data [12]. Moreover, feature extraction and classifier training are taken as two separate steps and not jointly optimized in most of these methods. All of these issues motivate us to explore new ways to improve the performance of activity recognition.

3 Proposed Method for Activity Recognition

3.1 Autoencoder

The autoencoder is a type of artificial neural networks that consist of three layers: input layer, hidden layer and output layer (see Fig. 1), with the constraint that the target values of the output layer are equal or approximate to the inputs during training. An autoencoder aims to learn a latent representation $h(x)$ of the input vector x. Suppose N and k denote the number of input units and the number of hidden units, respectively. Given a N-dimensional input vector x, the autoencoder transforms it to a latent representation $h(x) \in \mathrm{R}^k$ through a deterministic mapping (1),

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

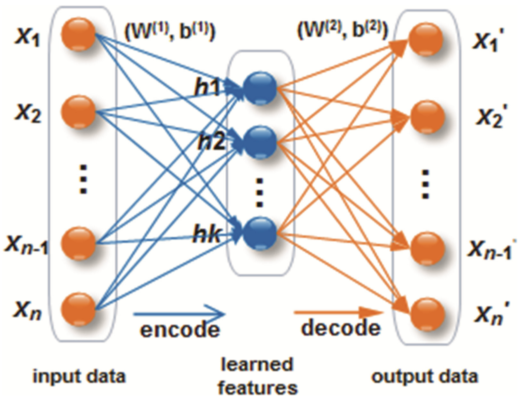

Fig. 1. The autoencoder architecture. The number of units in hidden layer is not necessarily less than that in the input layer.

where $W^{(1)} \in \mathrm{R}^{(k\mathrm{x}N)}$ is a matrix containing the weights from the input units to the hidden units, $b^{(1)}$ represents the bias of the hidden units, and $f(\cdot)$ is the activation function in each units. One of the most commonly used non-linear activation functions is sigmoid function shown mathematically as,

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

We then reconstruct the input x from the latent representation $h(x)$ using (3) and try to minimize the difference between x and x'.

$$x' = f(W^{(2)}h\big((x) + b^{(2)}\big), \tag{3}$$

where $W^{(2)} \in \mathrm{R}^{(N\mathrm{x}k)}$ contains the weights from the hidden units to the output units, and $b^{(2)}$ represents the bias of the output units. In such way, we can obtain a new feature representation $h(x)$ of x. Of note, the number of units in the hidden layers can be larger or less than the input dimension, enabling a larger exploration of non-linear relations.

With the aim to obtain a robust feature representation, Vincent et al. proposed the denoising autoencoders that try to reconstruct original data from a corrupted input with a local denoising criterion [25]. The corrupted inputs can be generated by adding random noises to the original inputs or randomly choosing a proportion of them and setting them to be zero. In this study, we use the denoising autoencoder as the building block of the proposed activity recognition model.

3.2 Stacked Autoencoder

Recent advances in deep learning show that a deep or hierarchical architecture can contribute to obtaining more complex and non-linear relations underlying in data when compared with these models with shallow structures that contain zero or only one hidden layer [26]. A stacked autoencoder (SAE) is such a hierarchy model, in which an autoencoder is a building block [27, 28]. In SAE, each layer is fully connected to its adjacent layer and there is no connection between units in each layer. In such architecture, the objective function of SAE is to reconstruct the inputs at the output layer. Similar to the autoencoder, each hidden layer of SAE is actually a high-level representation of the input. Interestingly, the number of units in a hidden layer can be equal to, larger or less than the input dimension. This enables us to sufficiently explore different high-level feature representations in a flexible way.

In training a stacked autoencoder, conventional gradient-based optimization methods, such as SGD and L-BFGS, suffer from the gradient diffusion and can easily be trapped into a poor local optimum on a network with randomly initialized weights and biases. To alleviate this problem and improve convergence rate, Hinton et al. proposed a greedy layer-wise learning process to learning a deep belief network and experimentally showed its good performance [27]. In such methods, we train each network separately rather than train them together, and the output of one network is the input of its following network. Specifically, we use the training data as inputs of an autoencoder to learn the first hidden layer, and then use the first hidden layer as input to learn the second hidden layer, and so on. Generally, assume that there is a stacked autoencoder with n layers and the first layer is the original data (training set). For the k-th autoencoder, $W^{(k)}$ are the weights from the input units to the hidden units, and $b^{(k)}$ are the biases of the 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 of interest. The above process is called *pre-training* because it works in an unsupervised way (without using corresponding labels).

3.3 Fine-Tuning the Activity Recognition Model

In order to perform activity recognition, the features learned in the stacked autoencoder are used with a set of labeled data to build a classifier. Accordingly, we can stack another output layer (classifier layer) on top of the SAE to classify an input. In this case, the feature vector encoded in the last hidden layer is the input of a learning algorithm in the classifier layer, and various classifiers are available for use. Figure 2 presents the overall architecture of an activity recognition model when the softmax classifier is used, in which the number of units in the classifier layer equals the number of activity classes.

To improve the performance of activity recognition, we further optimize the activity recognition model in a supervised manner. Specifically, we initialize the weights and biases of the deep network with values obtained in the *pre-training* process, and use the back propagation with gradient descent algorithm to optimize the model parameters. Prior researches show that such a strategy helps escape from the poor local optimum

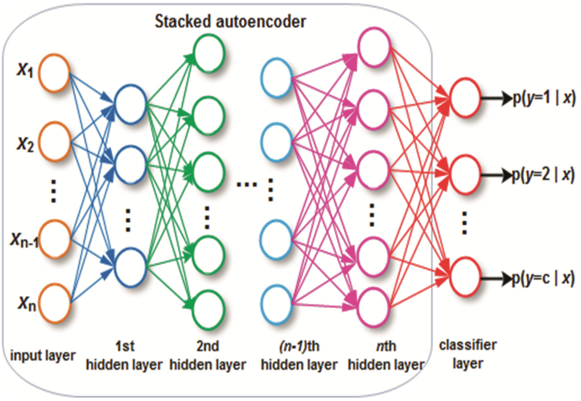

Fig. 2. Illustration to the activity recognition model with a stacked autoencoder and a softmax classifier. The last layer is the classifier layer, and the number of units equals to the number of activities of interest. The probability output determines the label of an input, where c indicates the c-th label. x_1, x_2,…, $x_{n\text{-}1}$ and x_n are a dimension of the original feature representation, 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.

and improve the time performance [28]. This procedure is called *fine-tuning* and works in a supervised manner (with labeled data involved).

To determine the optimal learning parameters and the network layout (e.g. how many hidden layers and the number of units in each hidden layer), besides the fine-tuning, we employ the grid search strategy and choose the best network structure as the final AR model via cross validation.

4 Experimental Results and Analysis

4.1 Experimental Datasets

To evaluate the performance of the proposed activity recognition model built on deep learning techniques, we conducted experiments on three publicly available datasets collected from three smart homes equipped with various simple sensors, respectively. Each of the smart homes housed one resident performing ADLs in it. For the first smart home (D1), there are three rooms equipped with fourteen sensors in total. Sensor data stream 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), thirteen activities were observed during a period of fourteen days in an apartment installed with twenty-three sensors. As a result, there are totally 200 activity instances consisting of 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. All information regarding the experimental dataset used in this study is briefly summarized in Table 1, and can be found in [29] for other details. Noticeably, all the sensors used are simple state-change sensors, including motion detector sensor, mercury contact, contact switch sensor, pressure mat, and float sensor. So, each dataset consists of binary temporal data that denote the activation of sensors.

Table 1. Experimental dataset description

Dataset	D1	D2	D3
Setting	Apartment	Apartment	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

4.2 Experimental Setup and Results

The sensor data stream was first divided into segments by shifting a fixed length, non-overlapping sliding window of sixty seconds as suggested by van Kastern et al. [21]. Then a N-dimensional feature vector $x_t = (x_t^1, x_t^2, \ldots, x_t^{N-1}, x_t^N)$ was extracted from each segment, in which N is the total number of sensors installed in a smart home and each dimension of x_t corresponds to a physical sensor. In our experiments, the original features of the sensor data can be represented in two forms: binary representation and numerical representation. The numerical representation method records the number of firings of a sensor during a specific time slice, while the binary representation method records whether a sensor fired at least once during the interval, and the value of a dimension is one if the corresponding sensor fired and zero otherwise. For the evaluation, we performed leave one day out cross validation, in which one full day of sensor data is used to test the classifier performance and sensor data of the remaining days are used for classifier training. We repeat the above process the number of days times and report the average results. Specifically, we evaluate the performance of the proposed model in terms of the following two metrics, the time-slice accuracy and the class accuracy, which can be calculated as follows.

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

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

where I(a == b) is the indicator function returning 1 if a equals b and 0 otherwise, M is the total number of sensor data segments in the test data, Nc denotes the number of segments belonging to class c, inferred(n) is the inferred label of segment n, and true(n) is the true label of segment n. In our study, rather than explore a large number of autoencoders, a two-layer stacked denoising autoencoder (SDAE) is used. Also, 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 the proposed activity recognition model with other four commonly used baselines, including Naïve bayes (NB), hidden markov model (HMM), 1-nearest-neighbor (KNN), and support vector machine with linear kernel (SVM). These four predictors with shallow structures directly use the binary representation and numerical representation rather than the learned high-level features as the inputs. For KNN, we use one nearest neighbor to decide the label of a test sample.

Tables 2, 3 and 4 present the experimental results on the three datasets, respectively. For each, 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. From Table 2, we observe that SDAE outperforms other four methods in terms of both time-slice accuracy and class accuracy whichever the original feature representation is adopted. Specifically, SADE obtained a time-slice accuracy of 85.32 % and a class accuracy of 49.91 % compared to the 59.11 % time-slice accuracy and 48.46 % class accuracy of the commonly used probability-based HMM in the case of binary representation. When using the numerical representation, SDAE achieved 85.52 % time-slice accuracy and 53.42 % class accuracy compared to the 59.73 % time-slice accuracy and 43.3 % class accuracy of HMM. For NB classifier, which is built on the basis of conditional independence among features, it performed poorly whichever feature representation was used. This indicates that there exist underlying relations among these features. Also, instance-based learning method KNN also failed to give good results, and consistently obtains the lowest time-slice accuracy and class accuracy. From Table 3, we can observe that deep learning based approaches outperformed their competing methods in time-slice accuracy. Although they failed to achieve the best class accuracy, their difference is quite small. For instance, SDAE obtained a class accuracy of 43.30 %, which was 1.21 % less than the best 44.51 % of SVM. Similar conclusions can be drawn from Table 4.

Table 2. Experimental results on dataset D1.

Method		NB	HMM	1NN	SVM	SDAE
Binary	Time-slice (%)	77.14	59.11	33.10	83.88	85.32
	Class (%)	42.62	45.48	32.43	48.14	49.91
Numerical	Time-slice (%)	77.03	59.73	33.30	83.95	85.52
	Class (%)	38.43	43.35	33.06	48.18	53.42

Table 3. Experimental results on dataset D2.

Method		NB	HMM	1NN	SVM	SDAE
Binary	Time-slice (%)	80.35	63.23	55.73	82.60	84.16
	Class (%)	32.47	44.66	30.47	41.76	39.92
Numerical	Time-slice (%)	80.50	66.79	59.03	81.82	85.61
	Class (%)	24.83	28.79	39.46	44.51	43.30

Table 4. Experimental results on dataset D3.

Method		NB	HMM	1NN	SVM	SDAE
Binary	Time-slice (%)	46.47	26.48	27.73	44.56	50.04
	Class (%)	16.84	17.22	19.81	21.33	21.14
Numerical	Time-slice (%)	41.44	27.37	30.82	42.70	54.82
	Class (%)	11.17	11.21	24.98	25.45	22.08

Overall, we can see that: (1) SDAE outperforms other four competing methods in terms of time-slice accuracy. In class accuracy, deep learning methods achieve better performance than NB, HMM, and 1NN when numerical representation is adopted, and obtain similar performance to others in the case of binary representation. (2) Deep learning based approaches are more robust to the choice of the original feature representation in comparison with other activity recognition models. For example, HMM obtained a class accuracy of 35.8 % in binary representation, decreased by 8.0 % compared to that of the numerical representation. This indicates that deep learning techniques generalize better across different original feature representations and can potentially relieve users of the reliance on domain knowledge to design and select features. Particularly, it should be noted that in this study, we do not fully explore the power of latent feature learning, since we only explore the deep learning architecture with two hidden layers and small number of units in each layer.

5 Conclusions

Wireless sensor network technology has great potential to be widely used in smart homes for human-centric applications due to its non-intrusiveness, low cost, and easy deployment. In activity recognition, researchers have conducted a wealth of work and proposed various models, while few explore how to learn useful features and to jointly optimize feature extraction and classifier construction. In this study, we present a deep learning based activity recognition model that uses an autoencoder to learn useful features from sensor data stream and unifies feature extraction and activity recognition in a single framework. To demonstrate the effectiveness of the proposed approach in activity recognition, we conducted experiments on three publicly available human activity recognition datasets and compared it with other four traditional methods in terms of time-slice accuracy and class accuracy. Experimental results show that our proposed method outperforms the competing methods, indicating its potential in human activity recognition. For the future work, we plan to further optimize the proposed model by varying the number of hidden layers and units in each layer, and study other feature learning methods.

Acknowledgments. This work was supported by the Natural Science Foundation of China (No. 61472057) and China Postdoctoral Science Foundation (No. 2016M592046).

References

1. Tapia, E.M., Intille, S.S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004)
2. Cook, D.J.: Learning setting-generalized activity models for smart spaces. IEEE Intell. Syst. **27**, 32–38 (2010)
3. Ordóñez, F., de Toledo, P., Sanchis, A.: Sensor-based Bayesian detection of anomalous living patterns in a home setting. Pers. Ubiquit. Comput. **19**, 259–270 (2015)
4. Suryadevara, N.K., Mukhopadhyay, S.C.: Determining wellness through an ambient assisted living environment. IEEE Intell. Syst. **29**, 30–37 (2014)
5. Liu, L., Peng, Y.X., Wang, S., Huang, Z.G., Liu, M.: Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors. Inf. Sci. **340–341**, 41–57 (2016)
6. Cook, D.J., Krishnan, N.C., Rashidi, P.: Activity discovery and activity recognition: a new partnership. IEEE Trans. Cybern. **43**, 820–828 (2013)
7. Krishnan, N.C., Cook, D.J.: Activity recognition on streaming sensor data. Pervasive Mob. Comput. **10**, 138–154 (2014)
8. Tapia, E., Intille, S., Haskell, W., Larson, K., Wright, J., King, A., Friedman, R.: 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. IEEE Press, New York (2007)
9. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Fox, D., Kautz, H., Hähnel, D.: Inferring activities from interactions with objects. IEEE Pervas. Comput. **3**, 50–57 (2004)
10. van Kasteren, T., Englebienne, G., Kröse, B.: An activity monitoring system for elderly care using generative and discriminative models. Pers. Ubiquit. Comput. **14**, 489–498 (2010)
11. Liu, L., Peng, Y.X., Huang, Z.G., Liu, M.: Sensor-based human activity recognition system with a multilayered model using time series shapelets. Knowl. Based Syst. **90**, 138–152 (2015)
12. Plötz, T., Hammerla, N.Y., Olivier, P.: Feature learning for activity recognition in ubiquitous computing. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 1729–1734. AAAI Press, California (2011)
13. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**, 1–127 (2009)
14. Chen, L., Hoey, J., Nugent, C.D., Cook, D.J., Yu, Z.: Sensor-based activity recognition. IEEE Trans. Syst. Man Cybern. Part C **42**, 790–808 (2012)
15. Figo, D., Diniz, P.C., Ferreira, D.R., Cardoso, J.M.: Preprocessing techniques for context recognition from accelerometer data. Pers. Ubiquit. Comput. **14**, 645–662 (2010)
16. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
17. Reiss, A., Hendeby, G., Stricker, D.: A competitive approach for human activity recognition on smartphones. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN, Belgium, pp. 455–460 (2013)
18. Wang, A.G., Chen, G.L., Yang, J., Zhao, S.H., Chang, C.Y.: A comparative study on human activity recognition using inertial sensors in a smartphone. IEEE Sens. J. **16**, 4566–4578 (2016)
19. Dernbach, S., Das, B., Krishnan, N.C., Thomas, B.L., Cook, D.J.: Simple and complex activity recognition through smart phones. In: 8th International Conference on Intelligent Environments, pp. 214–221. IEEE Press, New York (2012)
20. Kim, S.C., Jeong, Y.S., Park, S.O.: RFID-based indoor location tracking to ensure the safety of the elderly in smart home environments. Pers. Ubiquit. Comput. **17**, 1699–1707 (2013)

21. Van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. In Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 1–9, ACM Press, New York (2008)
22. Fleury, A., Vacher, M., Noury, N.: SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. IEEE Trans. Inf. Technol. Biol. **14**, 274–283 (2010)
23. Wilson, D.H., Atkeson, C.G.: Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 62–79. Springer, Heidelberg (2005)
24. Bhattacharya, S., Nurmi, P., Hammerla, N., Plötz, T.: Using unlabeled data in a sparse coding framework for human activity recognition. Pervasive Mob. Comput. **15**, 242–262 (2014)
25. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103. ACM Press, New York (2008)
26. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**, 504–507 (2006)
27. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
28. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems, pp. 153–160. MIT Press, Cambridge (2007)
29. van Kasteren, T., Englebienne, G., Kröse, J.A.B.: Human activity recognition from wireless sensor network data: benchmark and software. In: Chen, L., Nugent, C., Biswas, J., Hoey, J. (eds.) Activity Recognition in Pervasive Intelligent Environments. Atlantis Ambient and Pervasive Intelligence, vol. 4, pp. 165–186. Atlantis, Amsterdam (2011)