ORIGINAL RESEARCH

Evaluation of radial basis function neural network minimizing L-GEM for sensor-based activity recognition

Shuai Zhang[1] · Wing W. Y. Ng[2] · Jianjun Zhang[2] · Chris D. Nugent[1] · Naomi Irvine[1] · Ting Wang[2]

Abstract

Sensor-based activity recognition involves the automatic recognition of a user's activity in a smart environment using computational methods. The use of wearable devices and video-based approaches have attracted considerable interest in ubiquitous computing. Nevertheless, these methods have limitations such as issues with privacy invasion, ethics, comfort and obtrusiveness. Environmental sensors are an increasingly promising consideration in the ubiquitous computing domain for long-term monitoring, as these devices are non-invasive to inhabitants, yet certain challenges remain with activity recognition in sensorised environments, for example, addressing the challenge of intraclass variation between activities and reasoning from low-level uncertain information. In an effort to address these challenges, this paper proposes and evaluates the performance of a Radial Basis Function Neural Network approach for activity recognition with environmental sensors. The model is trained using the Localized Generalization Error and focuses on the generalization ability by considering both the training error and stochastic sensitivity measure. This measures the network output fluctuation with respect to the minor perturbation of input, to address the tolerance of the low-level uncertain sensor data. This approach is compared with three benchmark Neural Network approaches, including a popular deep learning approach using an Autoencoder, and it is evaluated with a simulated dataset as well as a number of publicly available datasets. The proposed method has shown advantages over the other models for all four evaluated datasets. This paper provides insights into the importance of model generalization abilities and an initial analysis of the limitation of deep Neural Networks with respect to sensor-based activity recognition.

Keywords Activity recognition · Uncertainty · Neural network · Localized generation error · Radial basis function neural network

✉ Shuai Zhang
  s.zhang@ulster.ac.uk

✉ Wing W. Y. Ng
  wingng@ieee.org

  Jianjun Zhang
  jjzhangscut@gamil.com

  Chris D. Nugent
  cd.nugent@ulster.ac.uk

  Naomi Irvine
  Irvine-N2@ulster.ac.uk

  Ting Wang
  1007376724@qq.com

[1] School of Computing, Ulster University, Newtownabbey, Co. Antrim BT37 0QB, Northern Ireland

[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

1 Introduction

Activity recognition serves as a key component of connected health, ambient assisted living and pervasive computing applications (Aggarwal et al. 2014; Espinilla et al. 2018), ranging from promoting physical activity to monitoring long term chronic conditions. It is a complex process that requires the deployment of sensors, data collection, and data modelling which is subsequently used to infer activities from the perceived sensor data (Chen et al. 2012). In this paper, we are mainly concerned with the modelling and perception of activities. Activity recognition is commonly used in rehabilitation systems for activity monitoring of inhabitants, and to support the management and also the prevention, of chronic disease. In relation to promoting physical activity, activity recognition is applied in rehabilitation centres that focus on stroke rehabilitation and those with motor disabilities (Chen et al. 2012). Another common application

domain for activity recognition is within smart homes, as a key motivation behind this research is to monitor the health of smart home inhabitants by tracking their daily activities. Activity recognition involves the automatic recognition of a user's activity in a smart environment using computational methods. These activities could be physical activities, i.e. standing and running as well as activities of daily living, i.e. dressing and preparing meals.

Sensor-based activity recognition has recently attracted considerable research interest in ubiquitous computing, predominantly due to advancements with wireless sensor networks and sensing technologies (Gu et al. 2011). Smart environments are an application of ubiquitous computing that rely on sensor data to perceive the environment, reasoning to assess how the environment could be changed, and actuators to make changes to it if required (Cook and Das 2007). The sensor activations capture user movement and interactions with objects in the environment, and therefore offer low-level but rich and fundamental information required for the recognition of human activities. There are challenges associated with activity recognition from such sensorised environments. For example, the sensor data readings could be unreliable (Ranganathan et al. 2004) due to hardware and communication issues such as sensor temporal malfunctioning and transmission error (Hong et al. 2009), and the collected data may not provide a full representation of the activities undertaken. Besides data quality, the challenge of intraclass variability requires consideration as an activity may be performed differently by various users and also by the same user at various times, which can affect activity modelling (Vogiatzaki 2015). Additionally, data collected may include sensor activations that are not representative of the current activity, due to human error or interleaved activities taking place.

Data-driven activity recognition approaches are therefore required to address the intraclass variation of activities and data uncertainty issues from the low-level information source. Neural Networks are non-parametric approaches that have the ability to implicitly detect complex nonlinear relationships between data and their classifications. Neural Networks have the potential to offer powerful modelling abilities for challenging problems, however, their application was partially restricted by computer computational capacities in earlier days. With the support of advances in computer hardware, it has enabled Neural networks to develop complicated architectures. Their state-of-the-art performance has recently attracted interest and attention in different research communities to address challenges in various application areas. Recently, there has been increased investigations into Neural Networks for sensor-based activity recognition, especially through the use of wearable and mobile devices (Wang et al. 2017a, b). Relatively, there has been less effort on exploring activity recognition

with Neural Networks, particularly with respect to activities of daily living carried out within smart environments.

In an effort to address the challenges of sensor-based activity recognition in smart environments, modelling approaches with high generalisation capacity to address the challenge of high intraclass variability within the same smart environment is therefore required. With the increasing popularity of the ambient living environment (Calvaresi et al. 2017), various projects have different hardware setup and data collection. The modelling approach should be applicable and effective to the different environment, in addition to addressing the unreliability of the low-level sensor information shared across different environment. This paper proposes and evaluates the performance of a Radial Basis Function Neural Network (RBFNN) approach for activity recognition with environmental sensors. The model is trained using the Localized Generalization Error Model (L-GEM). The proposed approach in this paper focuses on the generalisation of the model by considering both the training error and stochastic sensitivity measure. This is used to quantitatively measure the network output fluctuation with respect to the minor perturbation of network input, to address the uncertainty tolerance of low-level sensor data. Evaluations of the RBFNN are carried out on a number of simulated and publicly available datasets. The performance of the model is also compared against other popular Neural Network models, as well as a number of established classification methods. Given the recent popularity of deep Neural Network methods and their success in other application domains, such as image processing (Novotny 2014), computer vision (Ciresan et al. 2012; Bouchra et al. 2018), and natural language processing (Mikolov et al. 2013), this paper compares the performance of the RBFNN with an Autoencoder (Liou et al. 2014) which is amongst common approaches in deep learning. The major contributions of the proposed method include its fast training speed and high generalization capabilities compared with other neural network-based methods. The high performance achieved by the proposed method shows its effectiveness and robustness in sensor-based human activity recognition.

Related work on the methods and models for activity recognition is discussed in Sect. 2. The methodology of the proposed RBFNN via L-GEM is presented in Sect. 3 followed by its evaluation, comparison with other Neural Network approaches and discussion in Sect. 4. The paper concludes with future work and identified opportunities in activity recognition.

2 Related work

Approaches for the automatic recognition of activities are becoming a significant research area for application in smart environments and ambient assisted living scenarios and Internet of Things applications (García et al. 2017).

There has been extensive work in the literature on the activity recognition on the wearable sensors / devices (Hegde et al. 2017; Liu et al. 2017; Medina et al. 2017; Fullerton et al. 2017; Bulling et al. 2014), mostly focused on the physical activities such as running, sitting etc. These approaches are constrained with the participants wearing these devices and could be barrier to the uptake of long-term monitoring in a home environment. The other breath of work in activity recognition have been explored through the use of video-based approaches (Pirsiavash and Ramanan 2012; Rege et al. 2017; Jalal et al. 2017), which often require high computation costs. These methods, however, have limitations to consider such as issues with privacy invasion, ethics, comfort and obtrusiveness. In assisted living scenarios, for example, where activity monitoring occurs for elderly inhabitants, it has been reported that individuals are often reluctant to continuously wear body-worn sensors and are also reluctant to the installation of video-based monitoring due to privacy concerns (Roy et al. 2016). To avoid user acceptance issues and to address the concerns identified, binary sensors placed in the environment are an increasingly promising consideration in the ubiquitous computing domain for long-term monitoring, as these devices are non-invasive to inhabitants whilst also eliminating any privacy issues acknowledged with other approaches.

Binary sensors have been utilized in a recent study conducted by (Gochoo et al. 2017) to recognise four commonly performed Activities of Daily Living (ADLs) within a home monitoring environment. These activities include meal preparation, eating, relaxing and making a transition from bed to toilet. A Deep Convolutional Neural Network (DCNN) was implemented for the classification of these activities. The DCNN architecture consisted of two convolutional layers each followed by max-pooling layers, and subsequently two fully connected layers. The process involved converting the binary sensor data produced by 31 wireless passive infrared (PIR) motion sensors and 4 door sensors, into representative activity images for each of the activities defined. These images were then used to train and test the proposed DCNN classifier which produced an accuracy of 99.36% for ADL recognition. Although results produced are substantial, a larger number of activity classes could be investigated.

A recently conducted study (Moriya et al. 2017) used motion detectors attached to, or integrated within, various smart appliances to recognize activities of daily living. These appliances also included ON/OFF states for ceiling lights, IH cooking heaters, TV, PC, and cleaning appliances e.g. a vacuum, and OPEN/CLOSE states for appliances such as a kitchen fridge. Four participants performed nine activities within a smart home setting, which included activities such as sleeping, cooking and cleaning. A Random Forest model was chosen for activity classification, which achieved an accuracy of 68%. As future work has stated, this figure could be improved by applying more effective techniques and selecting effective features.

Smart home testbeds generated at the Center of Advanced Studies in Adaptive Systems (CASAS) that contain only passive, non-intrusive sensors have been used to test a deep belief network (DBN) implemented by (Fang and Hu 2014). Several activities that are considered difficult for elderly or disabled individuals to perform independently have been included in their study. The proposed DBN model was compared to other algorithms in terms of classification performance, with experimental results showing the DBN outperformed the Hidden Markov model and Naïve Bayes classifiers.

A stacked denoising autoencoder (SDAE) was implemented in (Wang et al. 2016) as an attempt to discover more intricate and non-linear relations for the classification of activity data acquired from numerous state-change binary sensors. The stacked autoencoder was first implemented for extracting features at a high-level, subsequently followed by the integration of a framework aimed at extracting relevant features and training the classifier. Evaluations of this method included testing the algorithm on three benchmark datasets and drawing performance comparisons against four well known classification models. Experiments revealed the proposed SDAE method outperformed other models comparatively in terms of recognition rate and the ability to generalize to unseen data. A limitation was stated in that the influence of latent feature learning was not fully explored during the study.

The inference of ADLs within a smart home setting makes use of an abundance of time-series data to achieve optimal feature extraction for activity classification in (Singh et al. 2017). Specifically, experiments included the implementation of convolutional (CNN) and recurrent (RNN) neural networks to classify activities such as sleeping, bathing and cooking. The RNN employed is a Long Short-Term Memory (LSTM) which is able to ascertain long-term dependencies within data, and the CNN employed is a one-dimensional temporal model consisting of four layers. Three benchmark datasets were used to evaluate model performance, which consisted of data acquired only through binary sensors including PIR motion sensors, pressure sensors, reed switches, and float sensors. The performances of the neural network models were compared to that of four common classifiers, with experimental results showing the LSTM outperformed all other models when tested against all three datasets considered in the study, followed by the CNN approach. Both neural network approaches performed significantly better than the other models.

Although deep learning models provide promising results in human activity recognition, major disadvantages have been identified, including the requirement of large amounts of high quality data and training time. Small amounts of

data may lead to insufficient training of deep learning models and poor generalization capabilities. The L-GEM Model method has demonstrated its effectiveness in supporting the development of classifiers, i.e. multi-layer perceptron (Yeung et al. 2016) and support vector machines (Sun et al. 2017), as well as its successful application in other domains, for example, feature selection (Ng et al. 2008) and sample selection (Ng et al. 2015). In order to achieve the minimized L-GEM function in this work, the selected RBFNN architecture is discussed for its application to activity recognition. To support the evaluation of the proposed method, we include several classification methods in the experiments. Experiments also include a deep learning stacked Autoencoder model, which is the most frequently used deep learning model for advanced feature representation using an unsupervised learning schema. In this way, the proposed method is compared with the most representative method, as well as other popular methods to demonstrate its effectiveness and robustness.

Despite previous effort in the literature on activity recognition approaches, this paper focuses on dealing with uncertainties of low-level environmental sensor data. It also focuses on evaluating the generalization capability of this approach for recognising a relatively large number of activities in a smart environment.

3 Methodology

This section is outlined as follows. The localized generalization error model is introduced in Sect. 3.1, followed by the Stochastic Sensitivity Measure and its analytical formula for RBFNN in Sect. 3.2. Finally, in Sect. 3.3, we describe the search method designed to discover the best architecture for RBFNN. The search method minimizes the L-GEM value of RBFNN and the network yielding the lowest L-GEM value will be selected.

3.1 Localized generalization error model

Using unseen samples very far away from training samples to evaluate the generalization capability of the classifier may be unmeaningful or misleading, as the classifier has never learnt knowledge about that region. Therefore, the localized generalization error model (L-GEM) has been proposed to provide an upper bound for the generalization error on the unseen samples, located within an identified small region of the training samples (Yeung et al. 2007). The L-GEM bounds above the training error for unseen samples. The training error of a classifier is defined by R_{emp} in Eq. (1):

$$R_{emp} = \frac{1}{N} \sum_{b=1}^{N} \left(F(x_b) - f(x_b) \right)^2 \tag{1}$$

where $F(x_b)$, $f(x_b)$ and N denote the target output on the training sample x_b, the real classifier output and the number of training samples in the dataset respectively.

For the purpose of evaluating the generalization capability of a classifier, in the L-GEM framework, samples located in the Q-neighborhood of x_b described in Eq. (2) are considered as unseen samples:

$$S_Q(x_b) = \left\{ x | x = x_b + \Delta x, |\Delta x_i| \le Q, i = 1, 2, \ldots, n \right\} \tag{2}$$

where n and Δx_i are feature numbers and the magnitude of perturbation of the ith input feature, respectively. Equation (2) shows that unseen samples are only allowed to deviate away from training samples no more than magnitude Q. The Q-union (S_Q) is the union of all Q-neighborhoods. The upper bound of the generalization error of a classifier for samples in the Q-union can now be computed by the L-GEM.

For a given Q, the L-GEM is given as follows in Eq. (3):

$$R_{SM}(Q) = \int_{S_Q} (F(x) - f(x))^2 p(x) dx \tag{3}$$

where $p(x)$ denotes the unknown probability density function of x in S_Q.

By applying Hoeffdings inequality with probability $1 - \eta$, we have Eq. (4):

$$R_{SM}(Q) \le \left(\sqrt{R_{emp}} + \sqrt{E_{S_Q}\left((\Delta y)^2 \right)} + A \right)^2 + \varepsilon = R_{SM}^*(Q) \tag{4}$$

where $\varepsilon = B\sqrt{\ln \eta / (-2m)}$, A, B, and $E_{S_Q}\left((\Delta y)^2 \right)$ denote the maximum desired output difference, the maximum possible value of the training error, and the stochastic sensitivity measure (ST-SM) of the output differences, respectively. In general, $A = B = 1$ holds for a classification problem with outputs ranging from [0, 1].

The ST-SM is then defined in Eq. (5) as the expectation of the squared differences between outputs of the training samples and unseen samples within their Q-neighborhood $(\Delta y = f(x_b + \Delta x) - f(x_b))$:

$$E_{S_Q}\left((\Delta y)^2 \right) = \frac{1}{m} \sum_{b=1}^{m} E\left[\left(f(x_b + \Delta x) - f(x_b) \right)^2 \right] \tag{5}$$

3.2 Stochastic sensitivity measure for RBFNN

The Radial Basis Function Neural Network (RBFNN) is employed in this work for activity recognition due to its efficient training speed and its capability of approximating a function with any precision rate given enough hidden neurons. An RBFNN can be described in Eq. (6)

$$f(x) = \sum_{j=1}^{M} w_j \exp\left(\frac{x - u_j^2}{-2v_j^2} \right) = \sum_{j=1}^{M} w_j \phi_j(x) \tag{6}$$

where M, w_j, u_j, and v_j denote the number of hidden neurons, the connection weight between the jth hidden neuron and the output neuron, the center vector and the width of the jth RBFNN hidden neuron, respectively.

The ST-SM quantitatively measures the output fluctuation of the neural network with respect to minor perturbation of the network input. In other words, the ST-SM measures if a network is sensitive to the input perturbation. Both the network inputs and connection weights could have their own mean and variance values (Yeung et al. 2007). Moreover, input and weight perturbations can be arbitrary. Thus, the perturbed samples can be treated as future unseen samples located around the training samples. In this work, we only consider the input perturbation and assume the inputs are independent and not identically distributed. The μ_{x_i} and $\sigma^2_{x_i}$ represent the expectation and variance of the ith input feature respectively. Without any prior knowledge, the input perturbation of the ith input feature is a random variable following a uniform distribution with zero mean and a variance of $\sigma^2_{\Delta x_i}$.

Let u_{ji} denote the ith input feature of the center of the jth hidden RBF neuron $\left(u_j = \left(u_{j1}, \ldots, u_{jn} \right)' \right)$, and $p(\Delta x)$ denote the probability density function of the input perturbations. Δx is uniformly distributed in the Q-neighborhood, i.e. $p(\Delta x) = 1/(2Q)^n$. For uniformly distributed input perturbations, we have $\sigma^2_{\Delta x_i} = \frac{(2Q)^2}{12} = Q^2/3$. Theoretically, we do not restrict the magnitudes of input perturbations as long as the variance of the input perturbation $\left(\sigma^2_{\Delta x_i} \right)$ is finite. Nevertheless, it is reasonable to assume uniform distribution here because all unseen samples should have an equal probability of occurrence without any prior knowledge on the distribution of unseen samples around the training samples.

By the law of large numbers, when the number of input features is not too low, $\phi_j(x)$ would have a log-normal distribution when n is not too small. Hence, the ST-SM of an RBFNN is given in Eq. (7) (Yeung et al. 2007):

$$E_{S_Q}\left((\Delta y)^2 \right) = \frac{1}{3}Q^2 \sum_{j=1}^{M} \gamma_j + \frac{0.2}{9}Q^4 n \sum_{j=1}^{M} \xi_j \tag{7}$$

where $\xi_i = \varphi_j / v_j^4$ and $\gamma_j = \varphi_j \left(\sum_{i=1}^{n} \left(\sigma^2_{x_i} + \left(\mu_{x_i} - u_{ji} \right)^2 \right) / v_j^4 \right)$. γ_j is defined by $\gamma_j = \varphi_j \left(\sum_{i=1}^{n} \left(\sigma^2_{x_i} + \left(\mu_{x_i} - u_{ji} \right)^2 \right) / v_j^4 \right)$. φ_j is defined by $\varphi_j = \left(w_j \right)^2 exp\left(\left(Var(s_j)/2v_j^4 \right) - \left(E(s_j)/v_j^2 \right) \right)$, where $E(\Delta)$ and $Var(\Delta)$ denotes the expectation operator and the variance operator, respectively, and s_j is given by $s_j = x - u_j^2$.

3.3 Finding optimal RBFNN using R^*_{SM}

RBFNN training aims to find a set of parameters that minimize the generalization error. A classic training method for RBFNN is that, by fixing the number of hidden neurons (M), the centers and widths are computed via the unsupervised k-means clustering method, and the connection weights are solved using the least square method. Therefore, RBFNN training aims to find an RBFNN with an optimal M value that minimizes L-GEM value (R^*_{SM}) among choices. In this section, a greedy technique based on R^*_{SM} is proposed to discover the optimal M value which makes use of the generalization capability of the RBFNN. The optimization problem is defined in Eq. 8 given the fix Q value:

$$\min R^*_{SM}(Q) \tag{8}$$

Given a training dataset with a given Q value, an RBFNN that yields a smaller R^*_{SM} value is preferable because it has higher generalization capability on unseen samples located within the Q-union. However, it is difficult to theoretically determine the Q value. A too large Q value may lead to a large R^*_{SM} value since too many dissimilar samples may be included in the calculation of the upper bound. Nevertheless, a too small Q value may lead to a Q-union containing too few unseen samples. In this case, one may consider revising the training data to include more of such data and retrain the classifier, since one may not expect a classifier to perfectly classify unseen samples that are totally different from the training data. As a rule of thumb, $Q = 0.1$ usually yields a good performance (Yeung et al. 2007), which means the maximum deviation from training samples is 10% for the input having been normalized to the range [0, 1].

The optimization problem (8) is solved by the simple greedy search algorithm (Zhang et al. 2017):

1. Start with M equals to the number of classes;
2. Train an RBFNN with M hidden neurons;
3. Compute the $R^*_{SM}(Q)$ value for the trained RBFNN;
4. If $M < N$, $M = M + 1$ and go to step 2.

4 Evaluation

The proposed Neural Network approach is compared with three popular Neural Network benchmarking approaches as well as a number of well-established machine learning methods, including a decision tree (CART), k-nearest neighbour (KNN), AdaBoost, Bagging, Naive Bayes, and Support Vector Machines (SVM) (Wu et al. 2008). The proposed method has also been compared with an RBFNN without LGEM to help clarify the usefulness of the minimization of LGEM for RBFNN training. The evaluation

has been carried out on a simulated dataset as well as a number of publicly available datasets.

4.1 Materials and methods

This section introduces three popular Neural Network approaches, namely, a deep learning method of a stacked autoencoder with softmax classifier, a Multi-Layer Perceptron Neural Network via minimized mean square error and the RBFNN without LGEM.

4.1.1 Autoencoder model

Deep Neural Networks aim to reveal distributed, high-level representations by utilizing hierarchical architectures. Generally, Convolutional Neural Networks (LeCun et al. 1998), Restricted Boltzmann Machines (Salakhutdinov and Hinton 2009) and Autoencoders (AE) (Liou et al. 2014) are the most commonly used in deep learning methods. Among them, the AE learns features from the original input as an unsupervised learning method (Baldi 2012). A deep architecture can be formed by stacking several AEs to improve the representation capability of the learned features. An AE consists of an input layer, an encoding layer, and a decoding layer. The encoding layer first maps an input x onto a hidden representation $f(x)$ through a deterministic mapping in Eq. (9):

$$f(x) = S_e\left(W_e x + b_e\right) \tag{9}$$

where W_e, b_e, and $S_e(\cdot)$ denote the weight matrix, the bias vector, and the activation function of the encoding layer respectively. Then, the encoding layer maps $f(x)$ back onto a reconstruction $g(f(x))$ of the same shape as x in Eq. (10):

$$g(f(x)) = S_d\left(W_e f(x) + b_d\right) \tag{10}$$

where W_d, b_d, and $S_d(\cdot)$ denote the weight matrix, the bias vector, and the activation function of the decoding layer, respectively. The aim of an autoencoder is to find a set of optimal parameters $\theta = \{W_e, b_e, W_d, b_d\}$ to minimize the reconstruction error between inputs x and outputs $g(f(x))$, formally represented in Eq. (11):

$$\arg \min_{\theta} \frac{1}{2} \sum_{i=1}^{N} x^{(i)} - g\left(f\left(x^{(i)}\right)\right)_2 \tag{11}$$

In the experiments, stacked autoencoders (SAEs) are utilised consisting of two AEs with the same activations to learn features. Figure 1 shows the work flow of the stacked autoencoder, and details of the feature learning algorithm for the SAEs can be found in Wang et al. (2017a, b).

4.1.2 MLP

The MLP method used in this work aims to find the best architecture for the Multi-Layer Perceptron Neural Network (MLPNN). We only consider the standard single hidden layer neural network and therefore the architecture here means the number of hidden neurons in the hidden layer. The MLPNN employed is trained using the off-the-shelf back-propagation method with the loss function being MSE. To find the best architecture, the MLP method utilizes a similar method as that of the RBFNN with L-GEM:

1. Start with M equals to the number of classes;
2. Train an MLPNN with M hidden neurons;
3. Compute the MSE value for the trained MLPNN;
4. If M < N, M = M + 1 and go to step 2.

The MLPNN with the smallest training MSE value is selected as the network with the best architecture.

4.1.3 RBFNN without L-GEM

The difference between the RBFNN with L-GEM and the RBFNN without L-GEM is how they find the best architecture (i.e. the number of hidden neurons). The RBFNN with L-GEM finds its best architecture via the greedy search method introduced in Sect. 3. However, the RBFNN without L-GEM finds its best architecture via the same search method, however with the goal being to minimise the training MSE of the network. The RBFNN with the smallest training MSE value is selected as the network with the best architecture.

4.1.4 Datasets

Four datasets have been used for the evaluation. These include the Kasteren Dataset (van Kasteren et al. 2008),
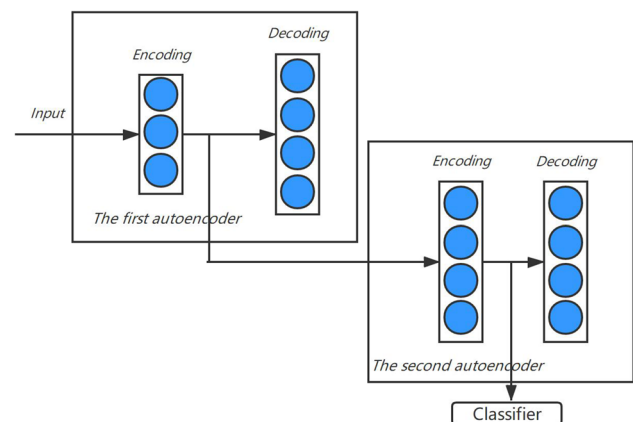

Fig. 1 Work flow of the stacked autoencoders with two hidden layers

Table 1 Evaluation datasets characteristics

Dataset	#Features	#Classes
kasterenADL	14	7
OrdonezA	12	9
OrdonezB	10	10
IESimData	21	11

OrdonezA and OrdonezB from the UCI ADL Binary Dataset (Ordycez et al. 2013) and the IESim Dataset (Synnott et al. 2014). The raw data were collected via the wireless sensor networks of various types of binary sensors including i.e., passive infrared (PIR), contact sensor, pressure sensors, depending on the projects experiments setup. The outputs of the sensors are binary where the value is 1 with the sensor being activated and 0 otherwise.

The characteristics of the datasets with respect to the number of features and the number of activities to be identified are summarised in Table 1.

The UCI ADL Binary dataset recorded ADLs performed by two users on a daily basis in their own homes. The ADLs were described by a set of sensors and the sensor events were captured by a wireless sensor network. The sensor events were recorded for 35 days in total, and the data was manually labelled. Two datasets have been obtained from this source, i.e. OrdonezA and OrdonezB. The OrdonezA contains 242 data points with 12 binary features and 9 activities. The OrdonezB contains 482 data points with 10 binary features and 10 activities.

The KasterenADL dataset recorded 7 ADLs performed by a 26-year-old man with 14 state-change sensors. The data was acquired over 28 days which resulted in 2120 sensor events and 242 activity instances.

IESim (Intelligent Environment Simulation) is a simulation tool which simulates the design and implementation of a real sensorized environment. Multiple sensors can be positioned on simulated objects and in the environment, and an avatar is used to represent the inhabitant. The simulation tool can be used to generate synthetic sensor datasets from the interactions of the avatar with the simulated smart environment.

Figure 2 shows the IESim environment used for data collection. Eight participants carried out eleven activities of daily living using the generated environment, including activities such as 'Go to bed', 'Watch TV' and 'Use Telephone'. Data collection resulted in 2231 sensor events and 308 activity instances. There were 21 sensors in total, represented in red asterisks in Fig. 2. Further details of data collection can be found in Synnott et al. (2016).

The metric employed to evaluate the model's performance is accuracy, which is the most commonly used metric. It describes the ratio of the number of correct predictions made
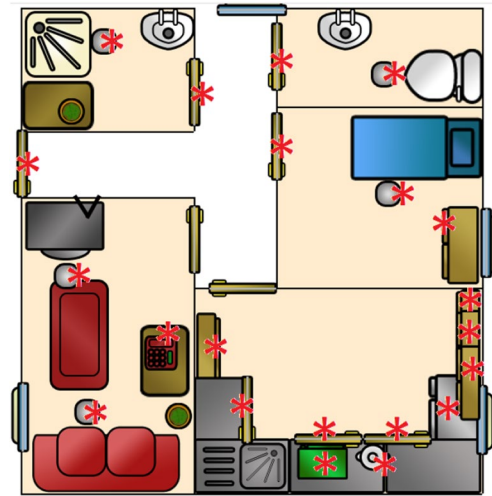

Fig. 2 the IESim environment with the sensor placements identified with red asterisks (Synnott et al. 2016)

by the model over the total number of test data instances. For the evaluation of the models, 10-fold cross-validation has been repeated five times to generate representative results.

4.2 Evaluation results and discussion

For evaluating the performance of the proposed RBFNN_LGEM method and conducting extensive research, we compared the proposed method with not only the neural networks mentioned in Sect. 4.1, but also several established classification methods, including a decision tree (CART), k-nearest neighbour (k-NN), AdaBoost, Bagging, Naive Bayes, and Support Vector Machines (SVM) (Wu et al. 2008). Table 2 shows that the RBFNN_LGEM yields the best performance in every experiment. The deep learning method (DNN) does not show advantage in comparison to traditional neural networks, even without minimizing the localized generalization error. DNNs usually perform best in image classification problems through finding nonlinear and local (convolutionary) feature representations among neighbouring pixels in images (Zeng et al. 2014). In contrast, the datasets used for sensor-based activity recognition consist of sensor data which focus more on the temporal relationships among sensor data. In addition to this, the signals need to be adapted to form virtual images for the DNN to process them, which may corrupt the correlations among consecutive signals. These may be the main reasons why the DNN does not yield good performance in sensor-based activity recognition. Both the DNN and the RBFNN use a linear classification (output) layer while the MLPNN uses a nonlinear classification (output) layer. Therefore, without the localized generalization error model, the MLPNN yields the best

Table 2 Comparison of classification accuracies of the models on the different data sources

Models	kasterenADL	OrdonezA	OrdonezB	IESim
CART	88.24 (0.76)	96.71 (0.03)	85.21 (0.23)	94.33 (0.3)
k-NN	88.14 (0.54)	97.38 (0.52)	83.34 (0.56)	95.24 (0.2)
AdaBoost	90.42 (0.4)	83.8 (0.54)	85.5 (0.21)	85.3 (0.27)
Bagging	89.12 (1.72)	85.39 (0.61)	85.5 (0.31)	86.93 (0.26)
Naïve Bayes	84.98 (0.46)	67.04 (0.59)	84.5 (0.16)	87.09 (1.2)
SVM	88.97 (0.61)	97.14 (0.03)	85.83 (0.25)	95.63 (0.18)
DNN (Stacked autoencoder)	86.74 (1.29)	73.98 (7.55)	68.76 (0.65)	82.59 (9.74)
MLPNN	85.01 (2.61)	94.86 (2.72)	83.19 (1.63)	93.05 (1.07)
RBFNN without LGEM	90.31 (1.32)	96.95 (0.45)	85.92 (0.44)	96.18 (0.43)
RBFNN_LGEM	90.91 (0.94)	97.51 (0.49)	86.00 (0.23)	96.44 (0.32)

All values are in percentage and that in the parenthesis are the standard deviation over five 10-fold Cross Validations

performances in three out of four experiments. When the RBFNN is optimized using the Localized Generalization Error, it yields the best performance. The RBFNN_LGEM merges the benefits of high generalization capability and fast training in comparison to both the MLPNN and the DNN. A classifier trained by minimizing the L-GEM can not only learn the training samples well by minimizing the training error, but can also avoid overfitting as it is not sensitive to input perturbations. Compared with the RBFNN without L-GEM, the RBFNN with L-GEM outperforms it in all four datasets, which shows the efficacy of the L-GEM. In comparison with the established classification methods, the proposed method also yields the best results in all four datasets, which demonstrates the robustness of the proposed method.

The Kasteren Dataset consists of sensor data generated from the same set of activities collected in different houses. This requires a higher level of generalization capability to yield a high accuracy. The RBFNN_LGEM outperforms the DNN, the MLPNN, and the RBFNN without L-GEM in the Kasteren Dataset by 4.81%, 6.94%, and 0.66%, respectively. These results show the RBFNN_LGEM yields outperformance than the other models,

demonstrating the importance of minimizing the Localized Generalization Error for neural network training.

All comparison methods are implemented using MatLab® Statistics and Machine Learning Toolbox. The main parameters settings for each method are given in the following. The maximum number of splits in CART is 20; the number of nearest neighbours in k-NN is 1; the number of learning cycles and the base learner in AdaBoost is 50 and discriminant analysis respectively; same parameters as that in AdaBoost are used in Bagging; Naive Bayes utilises the gaussian smoothing density estimate to model the data and SVM uses the gaussian kernel function and default values for the kernel are used.

In addition to evaluating the proposed method with regard to classification accuracy, the computational complexity of the proposed model has also been investigated. Tables 3 and 4 present the average time required in seconds for training and testing the models, respectively. Experiments are run using Matlab2017a under Windows 10 system on a computer with an intel i5-7300U CPU and 8 GB of RAM. For training, among all methods, the *k*-NN and the Naive Bayes required the least amount of time for models built from each of the datasets. The reason for this is the *k*-NN requires little

Table 3 Comparison of average training time (in seconds) of the models on the different data sources

Models	KasterenADL	OrdonezA	OrdonezB	IESim
CART	0.01	0.09	0.01	0.11
k-NN	0.01	0.08	0.01	0.08
AdaBoost	0.16	0.42	0.16	0.47
Bagging	0.13	0.32	0.13	0.34
Naïve Bayes	0.01	0.29	0.01	0.61
SVM	0.14	0.08	0.3	0.09
DNN (Stacked autoencoder)	3.51	3.75	3.56	3.94
MLPNN	0.29	0.46	0.39	0.39
RBFNN without LGEM	0.03	0.06	0.05	0.04
RBFNN_LGEM	0.03	0.06	0.06	0.04

Table 4 Average testing time (in seconds) of the models on the different test data sources. 0 for the entries represents that the time needed was very small

Models	kasterenADL	OrdonezA	OrdonezB	IESim
CART	0	0.01	0	0.01
k-NN	0	0.02	0	0.02
AdaBoost	0.02	0.09	0.02	0.1
Bagging	0.02	0.09	0.02	0.1
Naïve Bayes	0	0.06	0	0.12
SVM	0.02	0.01	0.03	0.01
DNN (Stacked autoencoder)	0.01	0.01	0.01	0.01
MLPNN	0.01	0.01	0.01	0.01
RBFNN without LGEM	0	0	0	0
RBFNN_LGEM	0	0	0	0

training but need to load all data into the RAM and "memorizes" them. The Naive Bayes method only requires fitting to a predefined distribution. Compared with the Neural Networks based methods, both RBFNN methods demand the least training time, especially in comparison to the Deep Neural Network Model. For the testing time presented in Table 4, both RBFNN methods require little time in comparison to the other methods. Based on the performance in prediction accuracy and model complexity, the proposed RBFNN_LGEM method offers fast training, testing, and high generalization capabilities. As a result, it has shown great potential in sensor-based human activity recognition.

Although some of the benchmarking datasets have been very well established in the research community, attention has been drawn to the limitations of publicly available datasets for activity recognition within smart environments. Data is usually collected in a controlled environment with limitations regarding the number of participants involved and the number of activities observed (Wang et al. 2018). There has been work attempting to address this issue in order to better support modelling and activity recognition using data collected from wearable devices (Cleland et al. 2014). However, there is limited progress on such data collection from environmental sensors for activity recognition.

5 Conclusion and future work

In this paper, we proposed a Radial Basis Function Neural Network approach trained using the Localized Generalization Error for the recognition of human activities within sensorised environments. This approach focused on generalization ability by considering both the training error and stochastic sensitivity measure, which measures the network output fluctuation with respect to the minor perturbation of input. This approach therefore deals with uncertainties in data from low-level sensor readings. In addition, this approach addressed the challenge

of intraclass variability where same activity may be performed differently by different individuals (Sun et al. 2017) as well as potential variations that may occur when the same individual performs an activity influenced by e.g. fatigue or stress (Cleland et al. 2018). To evaluate the proposed approach, a number of well-established public datasets have been used, as well as a dataset generated through a simulated environment. The proposed approach outperformed all benchmarking approaches used in this paper on all datasets, revealing the importance of model generalization abilities in sensor-based activity recognition.

In this work, raw data was used directly without any data pre-processing. One of our future works is to combine the LGEM-trained RBFNN with better features extracted from the raw sensor data to improve activity recognition performance. For instance, Word-to-Vector methods projecting a binary vector to a shorter real-valued or integer-valued vector may help with binary sensor data problems. On the other hand, owing to the simplicity of the binary sensor data, increasing the sampling rate to create a larger number of input features per time unit may help enhance feature representation. This will be helpful for real applications in which the user would collect their own data. For datasets with continuous sensor data, the window-size for an activity or sample is important. The optimal window-size can be learned through data using machine learning methods. Furthermore, the transition point from one activity to another is an important issue in sensor-based activity recognition. It would be interesting to explore the use of an RBFNN trained via the minimization of the Localized Generalization Error to optimize window-size and transition detection, in addition to activity recognition. We may also conduct research into a unified framework of Localized Generalization Error Minimization for all these tasks to perform activity recognition. Finally, regarding the dataset limitations discussed earlier, future evaluations of the proposed model could be carried out on a large-scale dataset acquired from a free-living environment.

References

Aggarwal JK, Xia L, Ann OC, Theng LB (2014) Human activity recognition: a review. Pattern Recogn Lett 48:70–80

Baldi P (2012) Autoencoders, unsupervised learning, and deep architectures. ICML Workshop of Unsupervised and Transfer Learning, 27: pp. 37–50

Bouchra N, Aouatif A, Mohammed N, Nabil H (2018) Deep belief network and auto-encoder for face classification. Int J Interact Multimedia Artif Intell. (in press)

Bulling A, Blanke U, Schiele B (2014) A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput Surv 46(3):33

Calvaresi D, Cesarini D, Sernani P, Marinoni M, Dragoni AF, Sturm A (2017) Exploring the ambient assisted living domain: a systematic review. J Ambient Intell Humaniz Comput 8(2):239–257

Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C Appl Rev 42(6):790–808

Ciresan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. Technical Report. arXiv preprint arXiv:1202.2745

Cleland I, Han M, Nugent C, Lee H, McClean S, Zhang S, Lee S (2014) Evaluation of prompted annotation of activity data recorded from a smart phone. Sensors 14(9):15861–15879

Cleland I, Donnelly M, Nugent C, Hallberg J, Espinilla M (2018) Collection of a diverse, naturalistic and annotated dataset for wearable activity recognition. In 2nd International Workshop on Annotation of useR Data for UbiquitOUs Systems (ARDUOUS 2018)

Cook DJ, Das SK (2007) How smart are our environments? An updated look at the state of the art. J Pervasive Mobile Comput 3(2):53–73

Espinilla M, Medina J, Hallberg J, Nugent C (2018) A new approach based on temporal sub-windows for online sensor-based activity recognition. J Ambient Intell Human Comput 1–13

Fang H, Hu C (2014) Recognising human activity in a smart home using deep learning algorithm. In: IEEE Control Conference, IEEE, pp. 4716–4720

Fullerton E, Heller B, Munoz-Organero M (2017) Recognising human activity in free-living using multiple body-worn accelerometers. IEEE Sens J 17(16):5290–5297

García CG, Núñez-Valdez ER, García-Díaz V, Bustelo CPG, Lovelle (2017) A review of artificial intelligence in the internet of things. Int J Interact Multimedia Artif Intell 4(3):7–10

Gochoo M, Tan T, Huang S et al (2017) DCNN-based elderly activity recognition using binary sensors. In: International conference on electrical and computing technologies and applications (ICECTA), IEEE, pp. 1–5

Gu T, Wang L, Wu Z, Tao X, Lu J (2011) A pattern mining approach to sensor-based human activity recognition. IEEE Trans Knowl Data Eng 23(9):1359–1372

Hegde N, Bries M, Swibas T et al (2017) Automatic recognition of activities of daily living utilizing insole based and wrist-worn wearable sensors. J Biomed Health Inf 22(4):979–988

Hong X, Nugent C, Mulvenna M, McClean S, Scotney B, Devlin S (2009) Evidential fusion of sensor data for activity recognition in smart homes. Pervasive Mobile Comput 5(3):236–252

Jalal A, Kim YH, Kim YJ, Kamal S, Kim D (2017) Robust human activity recognition from depth video using spatiotemporal multi-fused features. Pattern Recognit 61:295–308

LeCun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11): 2278–2324

Liou CY, Cheng WC, Liou J,W et al (2014) Autoencoder for words. Neurocomputing 139:84–96

Liu J, Sohn J, Kim S (2017) Classification of daily activities for the elderly using wearable sensors. J Healthcare Eng

Medina J, Fernandez-Olmo M, Pelaez M, Espinilla M (2017) Real-time monitoring in home-based cardiac rehabilitation using wrist-worn heart rate devices. Sensors 17(12):2892

Mikolov T, Sutskever I, Chen K et al (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119

Moriya K, Nakagawa E, Fujimoto M et al (2017) Daily Living Activity Recognition with Echonet Lite Applicances and Motion Sensors. In IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshop), IEEE, pp. 437–442

Ng WWY, Yeung DS, Firth M, Tsang ECC, Wang XZ (2008) Feature selection using localized generalization error for supervised classification problems using RBFNN. Pattern Recogn 41(12):3706–3719

Ng WWY, Hu JJ, Yeung DS, Yin SH, Roli F (2015) Diversified sensitivity-based undersampling for imbalance classification problems. IEEE Trans Cybern 45(11):2402–2412

Novotny D (2014) Large Scale Object Detection. Master Thesis, Czech Technical University, Prague

Ordycez FJ, de Toledo P, Sanchis A (2013) Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. Sensors 13:5460–5477

Pirsiavash H, Ramanan D (2012) Detecting activities of daily living in first-person camera views. In: Proceeding of the IEEE conference on computer vision and pattern recognition, IEEE, Providence, Rhode Island, pp 2847–2854

Ranganathan A, Al-Muhtadi J, Campbell RH (2004) Reasoning about uncertain contexts in pervasive computing environments. IEEE Pervasive Comput 3(2):62–70

Rege A, Mehra S, Vann A, Luo Z (2017) Vision-based approach to senior healthcare

Roy N, Misra A, Cook DJ (2016) Ambient and smartphone sensor assisted ADL recognition in multi-inhabitant smart environments. J Ambient Intell Humaniz Comput 7(1):1–19

Salakhutdinov R, Hinton GE (2009) Efficient learning of deep Boltzmann machines. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, Italy, pp. 693–700

Singh D, Merdivan E, Hanke S et al (2017) Convolutional and recurrent neural networks for activity recognition in smart environment. In: Towards integrative machine learning and knowledge extraction, Springer, pp. 194–209

Sun BB, Ng WWY, Chan PPK (2017) Improved sparse LSSVMs based on the localized generalization error model. Int J Mach Learn Cybernet 8(6):1853–1861

Synnott J, Chen L, Nugent CD, Moore G (2014) The creation of simulated activity datasets using a graphical intelligent environment simulation tool. In: Engineering in medicine and biology society (EMBC), Chicago, IL, USA, pp. 4143–4146

Synnott J, Nugent CD, Zhang S, Calzada A, Cleland I, Espinilla M, Quero JM, Lundstrom J (2016) Environment simulation for the promotion of the open data initiative. In IEEE international conference on smart computing (SMARTCOMP), St. Louis, Missouri, pp. 1–6

van Kasteren TLM, Noulas AK, Englebienne G, Kröse BJA (2008) Accurate activity recognition in a home setting. In: Proceedings of international conference on ubiquitous computing, pp. 1–9

Vogiatzaki E (2015) Modern stroke rehabilitation through e-health-based entertainment. 1st ed. Springer, Berlin pp 96–99

Wang A, Chen G, Shang C et al (2016) Human activity recognition in a smart home environment with stacked denoising autoencoders. In International conference on web-age information management, Springer, Cham, pp. 29–40

Wang T, Zeng GG, Ng WWY, Li JD (2017a) Dual denoising autoencoder features for imbalance classification problems. In: Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on, IEEE, pp. 312–317

Wang J, Chen Y, Hao S, Peng X, Hu L (2017b) Deep learning for sensor-based activity recognition: a survey. arXiv preprint arXiv:1707.03502

Wang J, Chen Y, Hao S et al (2018) Deep learning for sensor-based activity recognition: a survey. Pattern Recognit Lett

Wu X, Kumar V, Quinlan JR et al (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14(1):1–37

Yeung DS, Ng WWY, Wang DF, Tsang ECC, Wang XZ (2007) Localized generalization error model and its application to architecture selection for radial basis function neural network. IEEE Trans Neural Netw 18(5):1294–1305

Yeung DY, Li JC, Ng WWY, Chan PPK (2016) MLPNN training via a multiobjective optimization of training error and stochastic sensitivity. IEEE Trans Neural Netw Learn Syst 27(5):978–992

Zeng M, Nguyen LT, Yu B, Mengshoel OJ, Zhu J, Wu P, Zhang J (2014) Convolutional neural networks for human activity recognition using mobile sensors. In: 6th International conference on mobile computing, applications and services (MobiCASE), IEEE, pp. 197–205

Zhang S, Ng WWY, Zhang JJ, Nugent CD (20167) Human activity recognition using radial basis function neural network trained via a minimization of localized generalization error. In International Conference on Ubiquitous Computing and Ambient Intelligence, Springer, Cham, pp. 498–507