An Incremental Learning Method Based on Probabilistic Neural Networks and Adjustable Fuzzy Clustering for Human Activity Recognition by Using Wearable Sensors

Zhelong Wang, *Member, IEEE*, Ming Jiang, Yaohua Hu, and Hongyi Li, *Member, IEEE*

***Abstract*—Human activity recognition by using wearable sensors has gained tremendous interest in recent years among a range of health-related areas. To automatically recognize various human activities from wearable sensor data, many classification methods have been tried in prior studies, but most of them lack the incremental learning abilities. In this study, an incremental learning method is proposed for sensor-based human activity recognition. The proposed method is designed based on probabilistic neural networks and an adjustable fuzzy clustering algorithm. The proposed method may achieve the following features. 1) It can easily learn additional information from new training data to improve the recognition accuracy. 2) It can freely add new activities to be detected, as well as remove existing activities. 3) The updating process from new training data does not require previously used training data. An experiment was performed to collect realistic wearable sensor data from a range of activities of daily life. The experimental results showed that the proposed method achieved a good tradeoff between incremental learning ability and the recognition accuracy. The experimental results from comparison with other classification methods demonstrated the effectiveness of the proposed method further.**

***Index Terms*—Fuzzy clustering, human activity recognition, incremental learning, probabilistic neural networks, wearable sensor.**

I. Introduction

EXISTING researches have shown that human activity recognition by using wearable sensors could enable a range of health-related applications [1], [2]. A major goal of current studies on sensor-based human activity recognition (S-HAR) is to provide a long-term monitoring of daily activities of free-living subjects. Daily activities may provide additional information to medical doctors to accurately diagnose chronic diseases, as well as design the care plan of the patients [3]–[5]. Some studies on S-HAR focus on fall detection [6], [7], and some studies on S-HAR aim to estimate the energy expenditure during daily exercises [8], [9]. Furthermore, existing researches have shown that S-HAR could effectively improve the quality of healthcare provided to the elderly [10].

S-HAR from wearable sensor data is usually formulated as a classification problem [11]. Many classification methods have been used in previous studies. To name a few of them, Bao and Intille [12] employed decision table, instance-based learning, C4.5 decision tree, and naïve Bayes classifier to recognize 20 daily activities. Zhang *et al.* [13] used support vector machines (SVMs) and k-nearest neighbor (kNN) classification scheme to differentiate between falls and daily activities. Jamie *et al.* [14] used linear discriminate analysis and hidden Markov models to recognize nine workshop activities. Sun *et al.* [15] recognized human activities by means of artificial neural networks. Altun *et al.* [16] provided a comparative study on different techniques of classifying human activities, including Bayesian decision making, decision tree, least-squares method, kNN, dynamic time warping, SVMs, and artificial neural networks. Pärkkä *et al.* [17] used three classification methods to recognize everyday activities from wearable sensors, including custom decision tree, automatically generated decision tree, and artificial neural networks. Ermes *et al.* [18] detected daily activities by using a hybrid classifier combining a tree structure and artificial neural networks.

Most of the aforementioned classification methods lack the incremental learning ability. In many real-life scenarios, the monitoring of human activities may last for a long time. For example, the platform introduced in [19] was used to track physiological state of individuals for over four years, resulting 30 million minutes data with over 100 activities. In such a case, if a classifier used for S-HAR is only trained once from the sensor data collected during a few days or weeks, its accuracy and reliability are far from convincing. First, small-scale sensor data are usually not enough to include all the characteristics of selected human activities. An activity is usually performed differently by different persons, and even the same person may

Manuscript received August 1, 2011; revised November 11, 2011 and March 18, 2012; accepted April 12, 2012. Date of publication May 16, 2012; date of current version July 5, 2012. This work was supported in part by the China Postdoctoral Science Foundation under Grant 20080441102, the National Science and Technology Major Project under Grant 2010ZX04007-011-5, the China Earthquake Sector Research Funds under Grant 200808075, and the National Natural Science Foundation of China under Grant 61174027.

Z. Wang is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China, and also with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: wangzl@dlut.edu.cn).

M. Jiang was with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China. He is now with the School of Electronic Engineering, Dongguan University of Technology, Dongguan 523808, China (e-mail: onlyjiang@gmail.com).

Y. Hu is with the School of Electronic Engineering, Dongguan University of Technology, Dongguan 523808, China (e-mail: huyaohua@hotmail.com).

H. Li is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: hli@sia.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITB.2012.2196440

perform an activity differently at different conditions. Therefore, to include more characteristics, the training dataset of a classifier should be complemented from new sensor data collected from more persons and at more conditions. The classifier should be updated from the new training dataset to improve its classification accuracy. Second, the collected sensor data are probably be polluted by noise, baseline drift, sensor failure, and other factors. Because the accuracy of a classifier depends on the quality of training dataset, the polluted sensor data may decrease the classification accuracy. To avoid the risk that a classifier is trained from polluted sensor data, it is necessary to train a classifier repeatedly from many other datasets. Third, the activities which are expected to be recognized should be changed easily according to users' dynamic demand. Because during a long-term S-HAR mission, sometimes new activities are required to be detected, and sometimes existing activities are required to be removed. So, the classifier used for S-HAR should freely add new classes and remove existing classes to meet these dynamic requirements.

Several supervised neural networks with incremental learning ability have been developed in the past years. To name a few of them, incremental radial basis function (RBF) networks [20] are a new construction of RBF networks; its centers are dynamically inserted according to the accumulated error information of misclassifications, and the radius of each center is chosen based on neural gas algorithm; fuzzy ARTMAP (FAM) [21] is a variant of ARTMAP networks; ARTMAP is an incremental learning network based on adaptive resonance theory; FAM incorporates fuzzy set theory to govern the dynamics of ARTMAP; probabilistic FAM [22] is a hybrid utilization of FAM and probabilistic neural networks (PNNs) for online learning and prediction tasks; the use of PNN may improve the classification accuracy of FAM. The major drawbacks of these methods are that they are sensitive to the noise in training dataset and often suffer from overfitting problems, which may decrease their classification accuracy and generalization ability.

In this study, an incremental learning method is proposed for S-HAR. The proposed method is designed based on PNN and an adjustable fuzzy clustering algorithm (AFC). The proposed method has the following features. 1) It can easily learn additional information from new training samples to improve its recognition accuracy. 2) It can freely add new activities to be detected, as well as remove existing activities. 3) The updating process of the proposed method does not require previously used training data. Moreover, by differentiating the importance of pattern neurons in PNN, the robustness of the proposed method against noise in training dataset is strong. An experiment was performed to collect realistic wearable sensor data from a range of activities of daily life. The experimental results showed that the proposed method achieved satisfactory incremental learning ability, as well as high recognition accuracy. The organization of this paper is as follows: a short introduction of PNN and AFC algorithm is presented in Section II; a detailed introduction of the proposed method is presented in Section III; an experiment is described in Section IV; the experimental results are given in Section V; some discussions on the experimental results are presented in Section VI; and a conclusion is drawn in Section VII finally.

II. Theoretical Basis

A. PNNs

PNN is a classification method based on Parzen's theorem [23] for estimating the probability density function (pdf), as well as Bayesian decision theory [24] for decision making. The high computational efficiency and flexible structure of a PNN are quite suitable for S-HAR. The basic architecture of PNN is briefly described as follows. Detailed introduction of PNN can be found in [25].

1) *Input layer:* The input neuron does not perform any computation and just distributes the unlabeled sample $\boldsymbol{x}$ to the pattern layer.
2) *Pattern layer:* The number of pattern neurons is equal to the number of training samples. Each pattern neuron belongs to one class. Consider a classification task with M classes. Suppose there are K_m training samples whose labels are the mth ($m = 1, \ldots, M$) class, so there are K_m pattern neurons belonging to the mth class. Each pattern neuron first forms a dot product of $\boldsymbol{x}$ with a weight vector, and then performs a nonlinear operation on the dot product. The weight vector of the kth pattern neuron of the mth class is $\boldsymbol{w}_{m,k}$, where $\boldsymbol{w}_{m,k}$ is the kth training sample of the mth ($m = 1, \ldots, M$) class. Let $y_{m,k}$ denote the output of the kth pattern neuron of the mth class; $y_{m,k}$ is calculated by

$$y_{m,k} = \exp[(\boldsymbol{x} \cdot \boldsymbol{w}_{m,k} - 1)/\sigma^2] \tag{1}$$

where σ is the smoothing parameter. Assume that both $\boldsymbol{x}$ and $\boldsymbol{w}_{m,k}$ have already been normalized to unit length. In this study, all pattern neurons share a same smoothing parameter which is calculated by

$$\sigma = \frac{1}{M} \sum_{m=1}^{M} \frac{\sum_{i=1}^{K_m - 1} \sum_{j=i+1}^{K_m} \|\boldsymbol{w}_{m,i} - \boldsymbol{w}_{m,j}\|}{(K_m - 1)(K_m - 2)}. \tag{2}$$

3) *Summation layer:* The number of summation neurons is equal to the number of classes. The mth summation neuron averages the inputs from all the pattern neurons belonging to the mth class. Let z_m denote the output of the mth summation neuron; z_m is calculated by

$$z_m = \frac{1}{(2\pi)^{L/2}\, \sigma^L} \frac{1}{K_m} \sum_{k=1}^{K_m} y_{m,k} \tag{3}$$

where L is the dimension of training samples. According to Parzen's theorem [23], z_m is the estimation of pdf of the mth class.
4) *Decision layer:* There is one decision neuron deciding which class $\boldsymbol{x}$ should be classified to. The decision is based on Bayesian decision theory [24]. The output of the decision neuron is calculated by

$$output = \arg\max_{m} \{p_m z_m | m = 1, \ldots, M\} \tag{4}$$

where p_m is the prior probability of the mth class. In this study, all the classes share the same prior probability $p_m = 1/M$.

The major drawback of a traditional PNN is that all the training samples are used as weight vectors of pattern neurons. When the number of training samples is large, the computational efficiency of a PNN becomes quite low. The use of clustering in a combination with a PNN has been considered as a common way to reduce the computational load [26], [27]. Instead of original training samples, the center vectors of clusters are used to train a PNN.

B. Adjustable Fuzzy Clustering Algorithm

In this study, the AFC algorithm is utilized to group training samples into clusters. AFC is an extension of fuzzy c-means clustering algorithm (FCM). AFC exhibits a number of useful features which are associated with dynamic nature of underlying data. The use of AFC may give incremental learning ability to a traditional PNN. AFC is briefly described as follows. Detailed introduction of AFC can be found in [28]. The knowledge of FCM for understanding AFC can be found in [29].

Let Ω denote a set of training samples. First, AFC carries out FCM to group Ω into H clusters. Let the center vector of the hth ($h = 1, \ldots, H$) cluster be denoted by $\boldsymbol{\nu}_h$. Second, for each $\boldsymbol{\omega} \in \Omega$, a "degranulation" sample $\hat{\boldsymbol{\omega}}$ is calculated by

$$\hat{\boldsymbol{\omega}} = \sum_{h=1}^{H} u_h(\boldsymbol{\omega})^{\tau} \boldsymbol{\nu}_h \Big/ \sum_{h=1}^{H} u_h(\boldsymbol{\omega})^{\tau} \tag{5}$$

where $u_h(\boldsymbol{\omega})$ is the membership degree indicating with what degree $\boldsymbol{\omega}$ belongs to the hth cluster and τ is the fuzzy degree parameter. The degranulation sample $\hat{\boldsymbol{\omega}}$ is considered as an estimation of the optimum sample on a basis of $\boldsymbol{\nu}_h$ and $u_h(\boldsymbol{\omega})$. The distance between $\boldsymbol{\omega}$ and $\hat{\boldsymbol{\omega}}$ may constitute a viable assessment of the qualities of clusters. The reconstruction error of the hth cluster is calculated by

$$V_h = \sum_{\boldsymbol{\omega} \in \Omega_h} \|\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}\|^2 \tag{6}$$

where $\Omega_h = \{\boldsymbol{\omega} | \, h = \arg_i \min\{\|\boldsymbol{\omega} - \boldsymbol{\nu}_i\|\}, \boldsymbol{\omega} \in \Omega\}$. Third, AFC carries out splitting and merging mechanisms to the clusters to ensure $\max\{V_h\} < \varepsilon$, where ε is a predefined threshold indicating the maximum reconstruction error which can be tolerant.

1) *Splitting mechanism:* The splitting mechanism can split a cluster into two smaller clusters to lower the reconstruction error. If $\max\{V_h\} > \varepsilon$, AFC triggers splitting mechanism to the h_0th cluster, where $h_0 = \arg_h \max\{V_h\}$. The calculation of two new center vectors $\boldsymbol{\nu}_i^{\text{new}}$ $(i = 1, 2)$ are carried out iteratively according to

$$u_i^{\text{new}}(\boldsymbol{\omega}) = u_{h_0}(\boldsymbol{\omega}) \left[\sum_{j=1}^{2} \left(\|\boldsymbol{\omega} - \boldsymbol{\nu}_i^{\text{new}}\| / \|\boldsymbol{\omega} - \boldsymbol{\nu}_j^{\text{new}}\| \right)^{1/(\tau-1)} \right]^{-1} \tag{7}$$

$$\boldsymbol{\nu}_i^{\text{new}} = \sum_{\boldsymbol{\omega} \in \Omega_{h_0}} u_i^{\text{new}}(\boldsymbol{\omega})^{\tau} \boldsymbol{\omega} \Big/ \sum_{\boldsymbol{\omega} \in \Omega_{h_0}} u_i^{\text{new}}(\boldsymbol{\omega})^{\tau}. \tag{8}$$

Detailed derivations of (7) and (8) can be found in [30]. Then, AFC recalculates the reconstruction errors of the $H + 1$ clusters. If $\max\{V_h\} < \varepsilon$, the splitting mechanism is terminated. Otherwise, the splitting mechanism is repeated.

2) *Merging mechanism:* The merging mechanism can merge two clusters into a bigger cluster to reduce the number of similar clusters. If $\max\{V_h\} < \varepsilon$, AFC triggers merging mechanism to the h_1th and h_2th clusters, where $(h_1, h_2) = \arg_{(i,j)} \min\{\|\boldsymbol{\nu}_i - \boldsymbol{\nu}_j\| \, | i \neq j\}$. The new center vector $\boldsymbol{\nu}^{\text{new}}$ is calculated by

$$\boldsymbol{\nu}^{\text{new}} = \sum_{\boldsymbol{\omega} \in \Omega_{\text{new}}} u^{\text{new}}(\boldsymbol{\omega})^{\tau} \boldsymbol{\omega} \Big/ \sum_{\boldsymbol{\omega} \in \Omega_{\text{new}}} u^{\text{new}}(\boldsymbol{\omega})^{\tau} \tag{9}$$

where $\Omega_{\text{new}} = \Omega_{h_1} \cup \Omega_{h_2}$ and $u^{\text{new}}(\boldsymbol{\omega}) = u_{h_1}(\boldsymbol{\omega}) + u_{h_2}(\boldsymbol{\omega})$. Detailed derivation of (9) is presented in [28]. Then, AFC recalculates the reconstruction errors of the $H - 1$ clusters. If $\max\{V_h\} > \varepsilon$, AFC revokes the merging of these two clusters, and terminates the merging mechanism. Otherwise, the merging mechanism is repeated.

After carrying out the splitting and merging mechanisms, AFC acquires the optimal number of clusters from Ω, and limits the reconstruction errors below the predefined threshold ε. The center vectors of these clusters are used to train a PNN instead of original training samples.

III. Proposed Incremental Learning Method

The training and updating processes of the proposed method are illustrated in Fig. 1.

A. Preprocess Wearable Sensor Data

It is hard for a PNN to recognize human activities directly from original wearable sensor data without any preprocessing [11], as well as for other classification methods. A common way of preprocessing is feature extraction. Original wearable sensor data are first cut into small time segments as observation windows. Then, features are extracted from each window to characterize human activities. The label of a feature vector is the activity which is performed during the time of corresponding window. All labeled feature vectors are used as the training samples.

B. Establishing a New PNN From the First Training Dataset

Let S denote the first training dataset. First, the proposed method partitions S into M parts by activity labels, where M is the number of activities included in S. Let S_m $(m = 1, \ldots, M)$ denote the mth part, which consists of training samples whose labels are the mth activity. Second, the proposed method carries out AFC to each part. Suppose K_m clusters are acquired from S_m. Let the center vector of the kth ($k = 1, \ldots, K_m$) cluster of the mth activity be denoted by $\boldsymbol{w}_{m,k}$. Third, all center vectors are used to establish a new PNN according to Section II-A.

In the summation layer of a traditional PNN, the averaging process [shown in (3)] makes that all clusters are equally

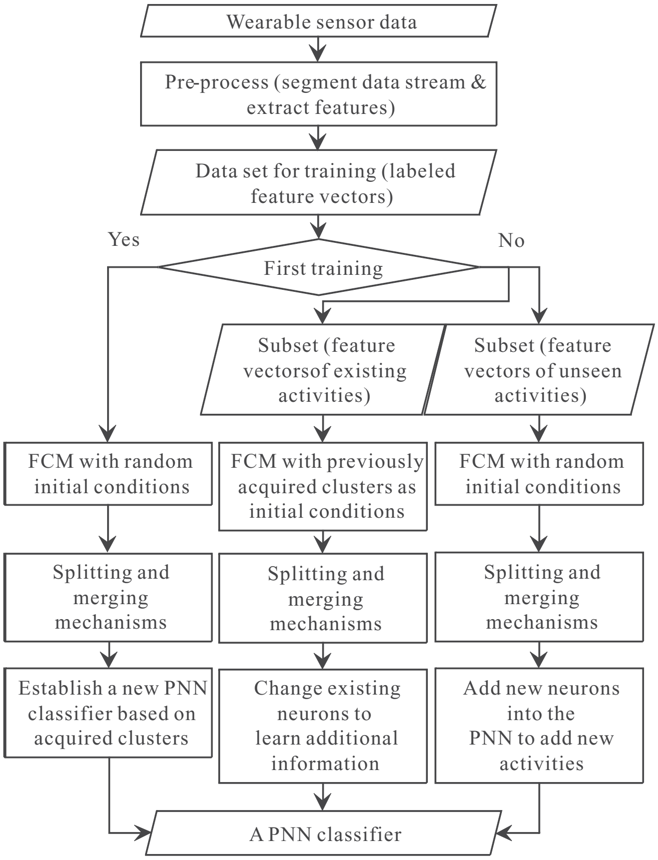


Fig. 1. Block diagram of the training and updating processes of the proposed method.

important for estimating the pdf. In the training dataset generated from realistic wearable sensor data, there are usually a few training samples that represent trivial or even wrong characteristics of human activities. The clusters from these interferential training samples may decrease the accuracy of a PNN. In this study, the importance of a cluster is measured as the number of training samples falling under the realm of this cluster. Of the K_m clusters of the mth activity, let the importance of the kth cluster be denoted by $\alpha(\boldsymbol{w}_{m,k})$. Let $I(\boldsymbol{w}_{m,k}, S_m)$ denote the number of training samples in $\{\boldsymbol{x}|k = \arg_i \min\{\|\boldsymbol{x} - \boldsymbol{w}_{m,i}\|\}, \boldsymbol{x} \in S_m\}$. When established a new PNN, let $\alpha(\boldsymbol{w}_{m,k}) = I(\boldsymbol{w}_{m,k}, S_m)$. In the summation layer of a PNN, the weighted averaging process is used to calculate the outputs of summation neurons. The output of the mth summation neuron is calculated by

$$z_m = \frac{1}{(2\pi)^{L/2}\sigma^L} \sum_{k=1}^{K_m} \left[\alpha(\boldsymbol{w}_{m,k}) \Big/ \sum_{i=1}^{K_m} \alpha(\boldsymbol{w}_{m,i}) \right]^{\tau} y_{m,k} \tag{10}$$

where the weight coefficient of the kth pattern neurons of the mth activity is $\alpha(\boldsymbol{w}_{m,k})/\sum_{i=1}^{K_m} \alpha(\boldsymbol{w}_{m,i})$.

C. Updating a PNN From New Training Dataset

Let S' denote a new training dataset. Suppose S' consists of training samples belonging to the previously appeared M activities. First, the proposed method partitions S' into M parts by activity labels. Let S'_m denote the mth part. Second, the proposed method carries out AFC to each part. At this time, the initial conditions of FCM algorithm are not randomly generated. For S'_m, the weight vectors of the existing K_m pattern neurons of the mth class are used as initial conditions of FCM. Suppose K'_m clusters are generated from S'_m. Let the center vector of the k'th ($k' = 1, \ldots, K'_m$) cluster of the mth activity be denoted by $\boldsymbol{w}'_{m,k'}$. Third, all existing K_m pattern neurons of the mth activity are replaced by K'_m new pattern neurons. The weight vector of the k'th new pattern neuron is $\boldsymbol{w}'_{m,k'}$. Though the previously used training dataset S_m is not directly used, the substantial knowledge acquired from S_m is still retained, because the center vectors of previously acquired clusters from S_m are used as the initial conditions of FCM to group S'_m [28].

Of the K'_m clusters of the mth activity, let $\alpha(\boldsymbol{w}'_{m,k'})$ denote the importance of the k'th cluster, and $\alpha(\boldsymbol{w}'_{m,k'})$ is calculated by

$$\alpha(\boldsymbol{w}'_{m,k'}) = I(\boldsymbol{w}'_{m,k'}, S'_m) + \sum_{k=1}^{K_m} \alpha(\boldsymbol{w}_{m,k}) \times \left[\sum_{i=1}^{K_m} \left(\frac{\|\boldsymbol{w}'_{m,k'} - \boldsymbol{w}_{m,k}\|}{\|\boldsymbol{w}'_{m,k'} - \boldsymbol{w}_{m,i}\|} \right)^{\frac{1}{\tau-1}} \right]^{-1}. \tag{11}$$

The first term in (11) indicates the importance of each cluster of the mth activity within S'_m. By using the second term in (11), the importance of previously acquired clusters from S_m is inherited.

D. Adding New Activities to be Detected and Removing Existing Activities

Let S'' denote a new training dataset that consists of training samples belonging to previously unseen activities. Without loss of generality, consider that there is only one new activity. First, the proposed method carries out AFC to S''. Supposes K'' clusters are generated. Let the center vector of the k''th ($k'' = 1, \ldots, K''$) cluster be denoted by $\boldsymbol{w}''_{k''}$. Second, the proposed method adds K'' new pattern neurons into the existing PNN. The weight vector of the k''th new pattern neuron is $\boldsymbol{w}''_{k''}$. A new summation neuron is also added to estimate the pdf of this new activity. The importance of the k''th new pattern neuron is denoted by $\alpha(\boldsymbol{w}''_{k''})$, where $\alpha(\boldsymbol{w}''_{k''}) = I(\boldsymbol{w}''_{k''}, S'')$. If an existing activity is decided not to be detected anymore, the proposed method just needs to remove the pattern neurons and the summation neuron of this undesired activity from the PNN.

IV. Experimental Design

A. Experiment Platform

To validate the effectiveness of the proposed method, an experiment was performed to collect realistic wearable sensor data from a range of daily activities. The experiment platform consisted of five collection nodes and a receiver node which are shown in Fig. 2. The collection node consisted of a sensor board, a wireless communication board, and a battery board.

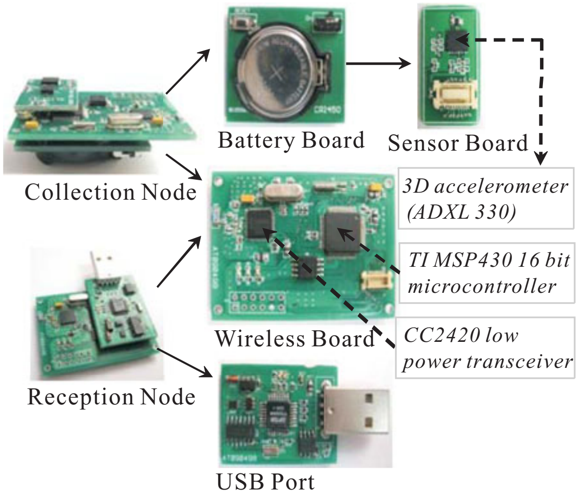
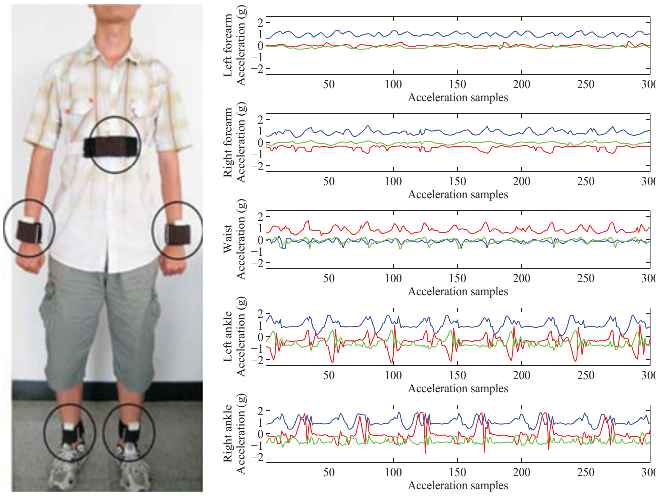


Fig. 2. Experiment platform consisting of five collection nodes and a receiver node.



(a) (b)

Fig. 3. (a) Subject wearing five collection nodes. (b) Period of acceleration data collected from "walking in a corridor."

The receiver node consisted of a wireless communication board and a USB port. The sensor board included a triaxial accelerometer (ADXL330) with a full-scale measuring range of ±3 g to objectively monitor human movements [31]. The wireless communication board ran TinyOS [32] on a microcontroller (MSP430). Wireless communication between the collection node and the receiver node was achieved through wireless transceiver chips (CC2420) with IEEE 802.15.4 protocol. A clock chip (PCF8583) with the measurement resolution of 0.01 s was integrated in each wireless communication board. The time information was included in the wireless data packets for synchronizing acceleration signals between five collection nodes. The receiving frequency of acceleration signals was set to be 20 Hz with the minimal packet loss rate.

B. Data Collection

The experiment was performed in a laboratory environment where clear definitions and accurate annotations of daily activities could be provided. Eight subjects (four males and four females, ages from 25–28) took part in this experiment. Each subject first wore five collection nodes on five body locations, including left forearm, right forearm, trunk, left ankle, and right ankle, as shown in Fig. 3(a). Then, each subject performed a range of daily activities which are listed in Table I. These activities were selected because they occurred frequently in everyday life. Fig. 3(b) shows a period of acceleration data collected from "walking in a corridor." The duration of each activity was 1 min. Each subject repeated an activity five times. There were totally 480 acceleration datasets (8 subjects × 12 activities × 5 times) collected in this experiment.

TABLE I
ACTIVITIES PERFORMED IN THIS EXPERIMENT

Number	Activity Description
1	Standing still
2	Sitting and relaxing on a couch
3	Sitting at a desk and using computer
4	Lying on a bed
5	Drinking or eating stuff
6	Walking in a corridor
7	Walking with holding a box in his/her arms
8	Walking upstairs or downstairs
9	Washing hands
10	Cleaning windows with a cleaning rag
11	Cleaning a table with a cleaning rag
12	Sweeping floor with a broom

V. EXPERIMENTAL RESULTS

A. Preprocess Acceleration Data

The loss rate of wireless data packets was low in the experiment, because of the large RF out power and low receiving frequency. Only a few sporadic acceleration signals were missed. In this study, a simple linear interpolation was implemented to compensate the missing signals. A sliding window technique was adopted to cut acceleration data into windows with a same length. The length of each window was 6 s. There was also a 50% overlap between adjacent windows to avoid information loss at the boundary of a single window. The extracted features from each window were mean, standard deviation, kurtosis, correlation between axes, and energy. The usefulness of these features has been demonstrated in prior studies [12]–[16]. All features were normalized to the interval [0,1]. The dimension of a feature vector was 75 (5 accelerometers × 3 axes × 5 features).

B. Predetermine Parameters of the Proposed Method

Four parameters are needed to be predetermined before carrying out the proposed method. Three were used for FCM, including fuzzy degree parameter, initial number of clusters, and number of iterations. More explanation of these parameters can be found in [29]. The other parameter was the threshold used for splitting and merging mechanisms. The optimal values of these parameters were chosen by using the intersectional search method. Each parameter was searched within an appropriate range. The searching range of fuzzy degree parameter was from 1 to 5; the searching range of initial number of clusters was from 1 to 150; the searching range of number of iterations was from 20 to 200; and the searching range of threshold was from 1 to 10. The boundaries of these searching ranges were decided based on empirical values. A number of comparative trials were performed. In each trial, four values were selected from the four searching ranges as the corresponding parameters. Then, the proposed method established a PNN by using these parameters, and tested the accuracy of the PNN. Among all comparative trials, the values with the highest accuracy were decided as the optimal parameters. To avoid that the validation of the proposed method was "testing on training data," the training data consisted of 96 acceleration datasets (eight per activity) which were randomly selected from all 480 datasets, and the testing data consisted of 48 acceleration datasets (four per activity) which were also randomly selected. Finally, the fuzzy degree parameter was set to be 2.7, the initial number of clusters was set to be 80, the number of iterations was set to be 150, and the threshold was set to be 1.9.

C. Learning Additional Information From New Training Data

In this section, the ability of "learning additional information from new training data to improve the recognition accuracy" of the proposed method was validated by using two cross-validation methods. The traditional PNN and the combination of PNN and FCM (short as "PNN + FCM") were also carried out to prove the effectiveness of the proposed method. All three methods were implemented under MATLAB environment.

1) *Fivefold cross validation:* The collected 480 acceleration datasets were randomly distributed into five unions denoted by Φ_i $(i = 1, \ldots, 5)$. Each union consisted of 96 acceleration datasets (eight per activity). A single union was retained for testing, and the remaining four unions were used for training. The validation was repeated five times with each union used exactly once for testing. The process of each validation consisted of four training and testing steps. Suppose Φ_1–Φ_4 were used for training and Φ_5 was used for testing. At the first step, the proposed method established a new PNN from Φ_1, and tested it from Φ_5. The traditional PNN and PNN + FCM were also trained from Φ_1 and tested from Φ_5. At the jth step $(j = 2, \ldots, 4)$, the proposed method updated the established PNN from Φ_j, and tested it from Φ_5. Because traditional PNN and PNN + FCM had no incremental learning ability, they were retrained from $\Phi_1 \cup \cdots \cup \Phi_j$ and tested from Φ_5. Fig. 4 shows the average experimental results at each step across five repeats of validation.

2) *Leave-one-out cross validation:* The collected 480 acceleration datasets were distributed into eight unions denoted by Ψ_p $(p = 1, \ldots, 8)$. The pth union consisted of all the acceleration datasets collected from the pth person. A single union was retained for testing, and the remaining seven unions were used for training. The validation was repeated eight times with each union used exactly once for testing. The process of each validation consisted of seven steps. Suppose Ψ_1–Ψ_7 were used for training and Ψ_8 was used for testing. At the first step, the proposed method established a new PNN from Ψ_1, and tested it from Ψ_8. The traditional PNN and PNN + FCM were also trained from Ψ_1 and tested fromΨ_8. At the jth step $(j = 2, \ldots, 7)$, the proposed method updated the established PNN from Ψ_j, and tested it from Ψ_8. The traditional PNN and PNN + FCM were retrained from $\Psi_1 \cup \cdots \cup \Psi_j$ and tested from Ψ_8. Fig. 5 shows the average experimental results at each step across eight repeats of validation.

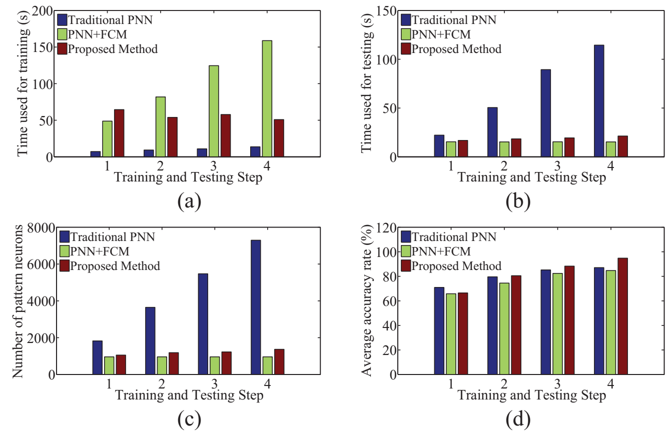


Fig. 4. (a)–(d) Average experimental results across five repeats of fivefold cross validation.

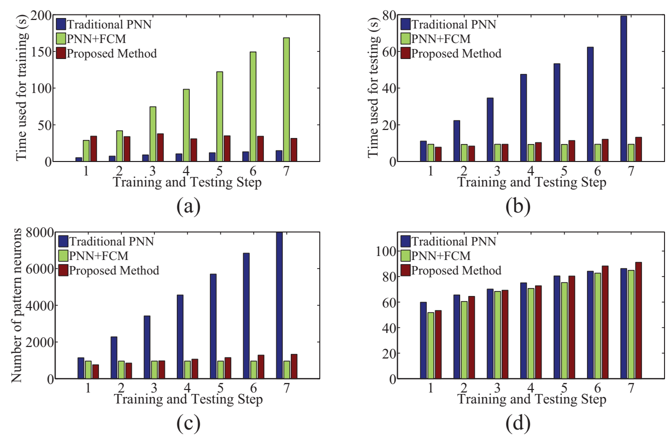


Fig. 5. (a)–(d) Average experimental results across eight repeats of leave-one-out cross validation.

As shown in Figs. 4(a) and 5(a), the training time of PNN + FCM increased rapidly along with the increase of the size of training dataset, and the training time of the proposed method was almost unchanged between different steps. As shown in Fig. 4(b) and (c) and Fig. 5(b) and (c), the number of pattern neurons and testing time of a traditional PNN increased rapidly along with the increase of the size of training dataset, and the number of pattern neurons and testing time of the proposed method were not directly impacted by the accumulation of training dataset. As shown in Figs. 4(d) and 5(d), the recognition accuracy of all three methods was low at the first step, and the proposed method achieved the highest recognition accuracy at the last step.

TABLE II
RECOGNITION ACCURACY OF EACH ACTIVITY AS ADDING NEW ACTIVITIES TO BE DETECTED

Step	Recognition accuracy of each activity (%)												Training
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Time (s)
1	95.7												33.2
2	95.7	94.4											32.5
3	95.6	94.3	90.5										29.8
4	95.7	94.3	90.5	90.6									33.6
5	95.6	94.2	90.5	90.7	90.1								27.8
6	95.7	93.2	89.8	89.4	90.1	89.4							30.7
7	95.6	93.3	89.7	89.3	89.8	89.2	95.4						26.4
8	95.6	93.3	89.7	89.3	89.8	89.2	95.3	91.3					30.3
9	95.5	93.1	89.3	89.2	89.6	87.8	95.3	91.2	88.2				31.1
10	95.4	93.1	89.4	89.3	89.7	87.9	94.7	91.0	88.3	92.2			32.1
11	95.3	93.1	89.4	89.1	89.5	87.8	94.5	90.0	88.3	92.0	91.4		34.6
12	95.4	93.0	89.2	89.2	89.5	87.6	94.1	90.8	87.5	92.0	91.2	93.7	29.6

TABLE III
RECOGNITION ACCURACY OF EACH ACTIVITY AS REMOVING EXISTING ACTIVITIES

Step	Recognition accuracy rate of each activity (%)												Training
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Time (s)
1	95.4	93.2	89.4	88.8	89.4	88.2	93.3	90.3	87.3	93.5	90.7	93.4	371.7
2	95.3	93.1	89.5	88.9	89.8	88.6	93.6	90.3	88.7	93.4	91.8		0.7
3	95.4	93.1	89.7	89.1	89.2	88.6	94.3	90.4	88.6	93.5			0.8
4	95.4	93.2	89.6	89.3	89.3	88.9	95.1	90.6	88.8				0.6
5	95.5	93.3	89.6	89.3	89.5	89.1	95.7	91.4					0.5
6	95.6	93.4	89.9	89.5	89.8	89.3	96.4						0.7
7	95.5	93.6	89.8	89.7	90.2	89.3							0.7
8	95.6	93.7	90.2	90.5	90.3								0.6
9	95.6	93.9	90.2	90.9									0.6
10	95.6	94.1	90.1										0.6
11	95.6	94.2											0.7
12	95.7												0.5

D. Adding New Activities to be Detected and Removing Existing Activities

In this section, the ability of "adding new activities to be detected and removing existing activities" of the proposed method was validated. Of the 40 acceleration datasets collected from the ath activity, 32 acceleration datasets were randomly selected and composed a union denoted by TR_a, and the remaining eight acceleration datasets composed a union denoted by TE_a. The proposed method was validated under MATLAB environment.

1) *Adding new activities to be detected:* The process of the validation consisted of 12 steps. At the first step, the proposed method established a new PNN from TR_1, and tested it from TE_1. At the kth ($k = 2, \ldots, 12$) step, the proposed method updated the established PNN from TR_k to add the kth activity to be detected, and then tested the PNN from $TE_1 \cup \cdots \cup TE_k$. The recognition results are listed in Table II. The symbols from "A1" to "A12" denote corresponding activities listed in Table I. As shown in Table II, the training time used for adding a new activity was short.
2) *Removing existing activities:* The process of the validation consisted of 12 steps. At the first step, the proposed method established a new PNN from $TR_1 \cup \cdots \cup TR_{12}$, and tested it from $TE_1 \cup \cdots \cup TE_{12}$. At the kth ($k = 2, \ldots, 12$) step, the proposed method removed the $(14 - k)$ th activity from the PNN, and tested the PNN from $TE_1 \cup \cdots \cup TE_{(13-k)}$. The recognition results are listed in Table III. As shown in Table III, the process of removing an existing activity almost needed no time, and the impact on the recognition accuracy of remaining activities was tiny.

E. Comparison With Other Classification Methods Used for S-HAR

To further validate the performance of the proposed method, six common classifiers and three incremental learning methods were carried out to recognize human activities from the collected acceleration data. The classifiers were naïve Bayes, C4.5 decision tree, decision table, multilayer perceptron (MLP), kNN, and SVM. These classifiers have already shown their effectiveness for S-HAR in prior studies [12]–[16]. Naïve Bayes, C4.5 decision tree, decision table, MLP, and kNN were implemented under WEKA environment [33], and SVM was implemented under MATLAB environment by using LIBSVM toolbox [34] with "one-against-one" approach for multiclass learning. The comparative incremental learning methods were increasing RBF networks, FAM, and probabilistic FAM. They were implemented under MATLAB environment.

The average recognition accuracy of the six common classifiers across 12 activities is listed in Table IV. As listed in Table IV, the proposed method achieved higher accuracy with both validation methods. The average recognition accuracy and the execution time of the three comparative incremental learning methods across 12 activities are listed in Table V. The execution time was measured as the sum of training time and testing time. As listed in Table V, the proposed method achieved both higher

TABLE IV
AVERAGE ACCURACY OF THE COMMON CLASSIFIERS ACROSS 12 ACTIVITIES

Method	5-fold	Leave-one-out
Naïve Bayes	82.3 %	77.5 %
C4.5	87.1 %	85.2 %
Decision table	76.7 %	73.3 %
MLP	90.2 %	87.4 %
kNN	85.9 %	81.7 %
SVM	87.8 %	85.1 %
Proposed method	91.3 %	89.2 %

TABLE V
AVERAGE ACCURACY AND EXECUTION TIME OF INCREMENTAL LEARNING METHODS ACROSS 12 ACTIVITIES

Method	5-fold		Leave-one-out	
	Accuracy	Time	Accuracy	Time
Increasing RBF	82.4%	175.6 s	80.5%	189.8 s
Fuzzy ARTMAP	86.4%	481.8 s	82.1%	527.9 s
Probabilistic FAM	87.6%	496.6 s	86.9%	544.3 s
Proposed method	91.3%	264.3 s	89.2%	325.4 s

accuracy than all three methods, and achieved shorter execution time than FAM and probabilistic FAM.

VI. DISCUSSION

The experimental results shown in Section V suggested satisfactory incremental learning ability of the proposed method. As shown in Figs. 4(a) and 5(a), to update an existing PNN from new training dataset, the training time of the proposed method was short. That was because only new training samples was used in the updating process, and the previously used training samples were not involved. Furthermore, as shown in Figs. 4(b) and 5(b), the number of pattern neurons of the proposed method did not increase along with the accumulation of training dataset. That was because by using the AFC, the number of pattern neurons was equal to the number of clusters which were dynamically adjusted according to the nature of training dataset. Because the testing time of a PNN was basically in proportion to the number of pattern neurons, the testing time of the proposed method also did not increase along with the accumulation of training data, as shown in Figs. 4(c) and 5(c). As shown in Figs. 4(d) and 5(d), the proposed method achieved the highest recognition accuracy at the last step of both validation methods. The recognition accuracy of a traditional PNN and PNN + FCM was partly decreased by the interferential training samples. The recognition accuracy of PNN + FCM was further limited by the constant number of pattern neurons. The proposed method overcame the weaknesses of a traditional PNN and PNN + FCM. By distinguishing the importance of clusters, the impact of interferential training samples was reduced. Furthermore, by using AFC to dynamically adjust the number of pattern neurons, additional information from new training samples could be learned. So, the ability of "learning additional information from new training data to improve the recognition accuracy" of the proposed method was proved adequately. The ability of "adding new activities to be detected and removing existing activities" of the proposed method was also proved by the experimental results. As shown in Tables II and III, to add or remove an activity, the training time of the proposed method was very short. That was because the process of adding or removing an activity was almost independent with the neurons of other activities. The proposed method just needed to add new neurons or remove existing neurons, and keep the other neurons unchanged.

The performance of the proposed method was further validated by comparing with other classification methods, including six common classifier and three incremental learning methods. As listed in Table IV, the proposed method achieved higher recognition accuracy than the six classifiers with both validation methods. The experimental results suggested that the proposed method offered good tradeoff between incremental learning ability and recognition accuracy. That is to say the proposed method not only has the incremental learning ability, but also could achieve comparable recognition results as other classifiers. As listed in Table V, the proposed method achieved higher recognition accuracy than the comparative incremental learning methods with both validation methods. Moreover, the execution time of the proposed method was much shorter than the execution time of FAM and probabilistic FAM. The recognition accuracy of comparative incremental learning methods was decreased by the interferential training samples and overfitting training.

It is worth noting that although the performance of the proposed method was just validated on wearable sensor data collected from normal subjects, the generalization capability of the proposed method was suggested to be satisfactory. That is because all experimental results were acquired by using subject-independent training, and the performance of the proposed method was good. Therefore, it could be concluded that the impact of different self-conditions of the subjects on the performance of the proposed method was limited. Of course, the viability of the proposed method still needs further validation from practical applications with the elderly and under clinical conditions.

VII. CONCLUSION

In this study, an incremental learning method is proposed to meet practical requirements of S-HAR. The proposed method was designed based on PNN and AFC. The proposed method can easily learn additional information from new training data to improve the recognition accuracy, as well as freely add new activities to be detected and remove existing activities. Furthermore, the updating process of the proposed method does not require previously used training data. The experimental results have shown that the proposed method achieved satisfactory incremental learning ability and recognition accuracy.

REFERENCES

[1] O. Aziz, B. Lo, J. Pansiot, L. Atallah, G. Z. Yang, and A. Darzi, "From computers to ubiquitous computing by 2010: Health care," *Phil. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.*, vol. 366, pp. 3805–3811, May 2009.

[2] D. McIlwraith and G. Z. Yang, "Body sensor networks for sport, wellbeing and health," in *Sensor Networks*, G. Ferari, Ed. Berlin, Germany: Springer-Verlag, 2009, pp. 349–381.

[3] C. Scanaill, S. Carew, P. Barralon, N. Noury, D. Lyons, and G. Lyons, "A review of approaches to mobility telemonitoring of the elderly in their living environment," *Ann. Biomed. Eng.*, vol. 34, no. 4, pp. 547–563, 2006.

[4] K. Chen and D. Bassett,Jr., "The technology of accelerometry-based activity monitors: current and future," *Med. Sci. Sports Exercise*, vol. 37, no. 11, pp. S490–S500, 2005.

[5] F. Pitta, M. Y. Takaki, N. H. Oliveira, T. J. P. Sant'Anna, A. D. Fontana, D. Kovelis, C. A. Camillo, V. S. Probst, and A. F. Brunetto, "Relationship between pulmonary function and physical activity in daily life in patients with COPD," *Respiratory Med.*, vol. 102, pp. 1203–1207, 2008.

[6] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Med. Eng. Phys.*, vol. 30, pp. 84–90, 2008.

[7] R. C. King, L. Atallah, C. Wang, F. Miskelly, and G. Z. Yang, "Elderly risk assessment of falls with BSN," in *Proc. 2010 Int. Conf. Body Sensor Networks*, 2010, pp. 30–35.

[8] J. Staudenmayer, D. Pober, S. Crouter, D. Bassett, and P. Freedson, "An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer," *J. Appl. Physiol.*, vol. 107, no. 4, pp. 1300–1307, 2009.

[9] S. J. Lin, L. Wang, B. Y. Huang, Y. T. Zhang, X. M. Wu, and J. P. Zhao, "A pilot study on BSN-based ubiquitous energy expenditure monitoring," in *Proc. 6th Int. Workshop Wearable Implantable Body Sensor Netw.*, 2009, pp. 49–52.

[10] L. Atallah, B. Lo, R. King, and G. Z. Yang, "Sensor placement for activity detection using wearable accelerometers," in *Proc. Int. Conf. Body Sensor Netw.*, 2010, pp. 24–29.

[11] S. Preece, J. Goulermas, L. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors—a review of classification techniques," *Physiol. Meas.*, vol. 30, pp. R1–R33, 2009.

[12] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. 2nd Int. Conf. Pervasive Comput. Commun.* (Lecture Notes in Computer Science, vol. 3001), 2004, pp. 1–17.

[13] T. Zhang, J. Wang, P. Liu, and J. Hou, "Fall detection by embedding an accelerometer in cellphone and using KFD algorithm," *Int. J. Comput. Sci. Netw. Security*, vol. 6, pp. 277–284, 2006.

[14] W. Jamie, P. Lukowicz, G. Troster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1553–1567, Oct. 2006.

[15] Z. Sun, X. Mao, W. Tian, and X. Zhang, "Activity classification and dead reckoning for pedestrian navigation with wearable sensors," *Meas. Sci. Technol.*, vol. 20, pp. 1–10, 2009.

[16] K. Altun, B. Barshan, and O. Tuncel, "Comparative study on classifying human activity with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, pp. 3605–3620, 2010.

[17] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 119–128, Jan. 2006.

[18] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 20–26, Jan. 2008.

[19] A. Teller, "A platform for wearable physiological computing," *Interact. Comput.*, vol. 16, pp. 917–937, 2004.

[20] B. Fritzke, "Fast learning with incremental RBF networks," *Neural Process. Lett.*, vol. 1, no. 1, pp. 2–5, 1994.

[21] G. A. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Netw.*, vol. 4, pp. 759–771, 1991.

[22] C. P. Lim and R. F. Harrison, "Probabilistic fuzzy artmap: An autonomous neural network architecture for Bayesian probability estimation," in *Proc. 4th Int. Conf. Artif. Neural Netw.*, vol. 409, pp. 148–153, Jun. 1995.

[23] E. Parzen, "On estimation of probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.

[24] A. M. Mood and F. A. Graybill, *Introduction to the Theory of Statistics*. New York: Macmillan, 1962.

[25] D. F. Specht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, pp. 109–118, 1990.

[26] P. Burrascano, "Learning vector quantization for the probabilistic neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 4, pp. 458–461, Jul. 1991.

[27] H. G. C. Traven, "A neural network approach to statistical pattern classification by 'semiparametic' estimation of probability density functions," *IEEE Trans. Neural Netw.*, vol. 2, no. 3, pp. 366–377, May 1991.

[28] W. Pedrycz, "A dynamic data granulation through adjustable fuzzy clustering," *Pattern Recognit. Lett.*, vol. 29, pp. 2059–2066, 2008.

[29] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[30] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 601–612, Jul. 1998.

[31] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiol. Meas.*, vol. 25, pp. R1–R20, 2004.

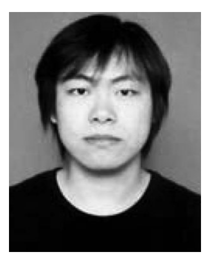
[32] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," *Ambient Intell.*, vol. 35, 2004.

[33] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[34] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–39, 2011.

Zhelong Wang (M'04) received the B.Sc. and M.Sc. degrees in automatic control from the Dalian University of Technology, Dalian, China, in 1996 and 1999, respectively, and the Ph.D. degree in robotics from the University of Durham, Durham, U.K., in 2003.

In 2004, he joined the School of Electronic and Information Engineering, Dalian University of Technology, where he is currently a Professor. He is also with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. His research interests include robotics, intelligent control, mechatronics, body sensor networks, and machine learning.



Ming Jiang received the B.Sc. degree in operations research and the M.Sc. and Ph.D. degrees in control theory from the Dalian University of Technology, Dalian, China, in 2006, 2008, and 2012, respectively.

In 2012, he joined the School of Electronic Engineering, Dongguan University of Technology, Dongguan, China. His research interests include body sensor networks, bioinformatics, and machine learning.

Yaohua Hu received the Ph.D. degree in automation of marine engineering from Dalian Maritime University, Dalian, China, in 1999.

During 1999–2006, he was an Associate Professor at Dalian Maritime University. In 2006, he joined the School of Electronic Engineering, Dongguan University of Technology, Dongguan, China, where he is currently a Professor. His research interests include predictive control and its application, ship control, very large scale integration design and verification, and viability theory.

Hongyi Li (M'04) received the B.Sc. degree from Nankai University, Tianjin, China, in 1996, the M.Sc. degree from the Chinese Academy of Sciences, Beijing, China, in 1999, and the M.Sc. and Ph.D. degrees from Harvard University, Cambridge, MA, in 2002 and 2004, respectively.

In 2007, he joined the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, where he is currently a Professor. His research interests include robotics and robotics control system.