Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-02229-y

ORIGINAL RESEARCH

Human activity recognition in smart environments employing margin setting algorithm

Ogbonna Michael Igwe[1] · Yi Wang[1] · George C. Giakos[1] · Jian Fu[2]

Received: 22 March 2019 / Accepted: 13 June 2020

Abstract
Human activity recognition is gaining promising attention in the research community with the recent revolution in artificial intelligence and machine learning to infer activities from the time-series sensor data. Significant progress has been made in the application of effective machine learning algorithms for pattern recognition and prediction of human activities in smart environments, such as ambient assisted living, healthcare monitoring, surveillance-based security and fitness tracking. In this paper, we propose to apply a supervised learning algorithm called margin setting algorithm (MSA) to predict the human activities. To validate the performance of MSA, we compare it with the support vector machine (SVM) and artificial neural network (ANN) and understand the activities of daily living of two residents in a smart home. The experimental results show that our proposed algorithm outperforms other state-of-the-art machine learning algorithms.

Keywords Margin setting algorithm · Machine learning · Activity recognition · Supervised learning · Smart environment

1 Introduction

Human activity recognition grabbed considerable research attentions because of its prominence in smart environments using IoT (Internet of Things), such as smart cities, smart homes, smart commercial buildings and factories, smart vehicles, smart energy, etc. A smart environment can be considered as an advanced automated system that forms an interconnection between the sensors, actuators and other computing devices. These devices are embedded in everyday human devices to make life more convenient and comfortable. Its eminence can be attributed to the advancement in sensor technology, data collection and management and more efficient algorithms. These advances have further pushed the supporting technologies from mere data gathering and communication to a more improved method of data gathering, processing, activity recognition. These advances are shaping the world in different ways. Smart homes employ human activity recognition system can watch the elderly or physically impaired people consistently, and predict their future activities. This improves the quality of life for the elderly and provides relief for family caregivers (Hassan et al. 2018; Liouane et al. 2018). Surveillance-based security system tends to use activity learning, monitoring and recognition to address abnormal behaviors and detect threats (Dahmen et al. 2017). Smart hospitals use activity recognition to improve patient's healthcare, experience, improve treatment and personalize assistance (Holzinger et al. 2015). To employ activity recognition, we may need to employ the following steps: (1) Choose appropriate sensor to monitor actor's movement and environmental changes (2) use appropriate data collection and analysis tool for data gathering, processing and storage (3) employ appropriate activity recognition computational model for easy software integration and manipulation (4) employ appropriate activity recognition algorithm to infer activities from sensor data. The type of sensors used for activity recognition can be classified as sensor-based, object-based and vision-based.

✉ Ogbonna Michael Igwe
oigwe01@manhattan.edu

✉ Yi Wang
yi.wang@manhattan.edu

George C. Giakos
george.giakos@manhattan.edu

Jian Fu
jian.fu@aamu.edu

[1] Electrical and Computer Engineering Department, Manhattan College, Riverdale, NY 10471, USA

[2] Electrical Engineering and Computer Science Department, Alabama A&M University, Normal, AL 35762, USA

Published online: 21 June 2020

Sensor based activity recognition has been in existence since the 1990s. It tends to employ sensor network and technology to gather actor's behavioral and environmental data. Early work involved wearable devices which are attached to the actor's body and mobile phones. This was due to their portability, low cost and ease of use, which later led to extensive works in context-awareness (Sezer et al. 2017), internet appliances (Saha et al. 2017), activity detection and classification (Cornacchia et al. 2016; Van Laerhoven et al. 2001). Wearable sensors attached to the actor include accelerometer for body movement, chest electrode for electrocardiogram (ECG), RFID (radio frequency identification) for tagging, phonocardiograph for heart sound, to name a few. The collected data is analyzed using machine learning or data mining techniques to discover trends or patterns in data. Wearable sensors mostly monitor human physical movements such as walking, standing and running. This method is effective but has a couple of limitations in the real world. Firstly, wearable sensors can prove uncomfortable to wear due to its size. It also suffers from issues such as battery life, ease of use and its applicability in the real world. Secondly, when dealing with issues that involves multiple motion movements or multiple activities with the environment, wearable sensors have proven ineffective, for example making pasta or rice. Due to some of these limitations, researchers have proposed a more suitable way that is based on user-object interaction. This birthed the introduction of object-based activity recognition discussed below.

Object based activity recognition also known as dense sensing-based activity is the practice of learning human activities by placing sensors on objects to monitor human interaction and behavior (Chen and Nugent 2019). Researchers hypothesized that activities can be recognized by manipulating the objects the interact with. In other words, by attaching sensors in an environment to monitor human activities and interaction, the activities can be recognized and predicted in the sensor data. It is receiving continuous attention in the research community because of advancements in cost-effective, intelligent and low-energy consumption sensors which are becoming readily affordable. Its advancement has led to lots of research in ambient assisted living (AAL) i.e. the use of intelligent systems embedded in intelligent objects to enable mostly elderly people live independently while monitored (Palumbo et al. 2016). Indication of an attached sensor to an object indicates a performed activity. From the sensor data and observation, we can infer what activity is performed thereby providing data on performed ADL. Despite the advancements and effectiveness of this method, it still faces some drawbacks. While object-based activity recognition has proved effective in single-resident activity recognition, it has proved ineffective for multiple resident activity recognition (Cook et al. 2013; Mohamed et al. 2017). Due to some of these drawbacks, researchers have turned to vision-based activity recognition.

Vision-based activity recognition makes use of cameras for surveillance, security monitoring, robotics, diagnosis (Zhang et al. 2017). It has gained prominence in computer vision techniques, human–computer interaction, user-interface interaction and surveillance because of its ability to analyze patterns and trends through visual observation. Researchers have used a variety of tools such as cameras, infra-red, multi-camera to model trends in ADL. To analyze vision-based activity recognition activities, the following must be computed: (1) a need for accurate object or human detection of the subject (2) a need for behavioral tracking and targeting (3) accurate activity recognition and finally (4) activity assessment. Despite the successes recorded by vision-based activity recognition, some of its limitations includes its invasiveness in privacy (some actors are not comfortable with cameras placed in some certain areas especially when dealing with smart home environments), difficulty in processing an activity captured from multiple cameras and human/object activity tracking and identification.

The aim of this work is to propose a new application of the Margin Setting Algorithm (MSA), to recognize and predicting user activities for object-based activity recognition. Based on the activity recognized by this method, appropriate service can be provided to user automatically in smart environments. We recently applied MSA to study its effectiveness for human activity recognition and yielded effective results. However, it has several limitations (Igwe et al. 2018). This work is an extension of the previous study with the following significant improvements:

(1) An improved explanation of MSA algorithm for activity recognition;
(2) A comparison analysis of the proposed algorithm to other popular algorithms in the experimental study;
(3) Time complexity of MSA algorithm is discussed and a Big-O notation is derived.

To the best of our knowledge, this is the first comprehensive study that applies MSA to a human activity recognition with a more rigorous and in-depth analysis of MSA.

The rest of the paper is structured as follows: Sect. 2 focuses on previous works done on human activity recognition. We focus on some of the findings, theoretical and methodological contributions using machine learning techniques. Section 3 discusses the methodology and algorithm of MSA. In Sect. 4, we present the comparison analysis of the proposed MSA with the support vector machine (SVM) and artificial neural network (ANN). Section 5 presents the summary and conclusion of this work.

2 Related work

Recent research that focus on human ADL using sensors have been tackled using mostly the generative and discriminative modelling approach. A summary of the approaches is shown in Table 1. The generative modeling approach tries to use a probabilistic method to build a perfect description of an input model. One of the limitation of these methods is its use of large dataset to learn every probabilistic outlook that may be required for optimum results. Examples include Hidden Markov Model (HMM), Naïve Bayes, and Bayesian network. HMM is an effective algorithm in modelling time series data. It uses a probabilistic approach to learn from data and make predictions from that data. It has become quite popular in the generative approach, and has found numerous applications in speech recognition, molecular biology, data compression, computer vision and numerous areas of artificial intelligence and pattern recognition (Malaisé et al. 2018). The Bayesian network is another generative approach that uses a probability approach to model a set of variables and their conditional dependencies. Liu et al. used Bayesian network-based probabilistic generative framework to recognize human activities. The benefit of this framework is using a primitive event-based methodology that can recognize complex activities to be shared across a wide range of sensor types (Liu et al. 2017).

Alternatively, the discriminative modeling approach tries to map an input variable to an output outcome. It tends to consider every possible input, then applies a classification boundary or margin to infer an output. Examples include Fuzzy logic, ANN, SVM, etc. Chen et al. presented a method that utilizes dynamic linear discriminant analysis (LDA) and fuzzy basis function (FBF) classifiers to recognize human daily activities from tri-axial accelerometer. However, it failed to recognize a wide range of daily activities accurately (Chen et al. 2008). Medina-Quero et al. combined fuzzy logic with recurrent neural networks to for real time activity recognition. Specifically, the proposed multiple Fuzzy Temporal Windows (FTW) collected the binary sensor activations and improved the recognition performance (Medina-Quero et al. 2018). Batchulunn et al. proposed a noel fuzzy system based behavior recognition that recognized eleven human activities in surveillance environments. The advantage of this method is twofold. First, this method can output plural recognition results at the same time, i.e., a person may wave his hand while walking. Second, it makes use of the previous recognized activities for activity recognition. This results in less error in noisy environment (Batchuluun et al. 2017). However, some other activities were not analyzed, such as sleeping and fainting.

ANN techniques have been successfully applied to various human activities recognition. Its success can be attributed in its ability to learn and map complex nonlinear separable features in data and make effective predictions. The decision boundaries are defined by the weights, which are mapped to some feature space. The multilayer feedforward neural network (FNN) was adopted to classify human static activities (e.g. standing and sitting) from human dynamic activities (e.g. walking and running) from

Table 1 Recent human activities recognition approaches

	Methods	Datasets used
Generative modeling approach	Hidden Markov Model Malaisé et al. (2018)	Xsens MVN suit Roetenberg et al. (2009) e-glove from Emphasis Telematics (Emphasis Telematics)
	Bayesian network Liu et al. (2017)	OSUPEL dataset Brendel et al. (2011) Opportunity dataset Roggen (2010) CAD14 Lillo et al. (2014)
Discriminative modelling approach	Fuzzy basis function Chen et al. (2008)	MMA7260Q tri-axial accelerometer, Freescale® Semiconductor
	Fuzzy logic + recurrent neural networks Medina-Quero et al. (2018)	Ordoñez dataset Ordóñez and Roggen (2016) CASAS dataset Cook et al. (2012)
	Fuzzy system Batchuluun et al. (2017)	Weizmann database (Blank et al. 2005)
	Feedforward neural networks Yang et al. (2008)	MMA7260Q tri-axial accelerometer, Freescale® semiconductor
	Deep Convolutional Neural Network Ronao and Cho (2016)	Accelerometer and gyroscope tri-axial sensor from smartphones
	Deep belief network Hassan et al. (2018)	human activities and postural transitions dataset Reyes-Ortiz et al. (2016)
	One-class SVM Yin et al. (2008)	Two-axis accelerometer, magnetometer from MICA2s (MPR400)
	Multi-class SVM with binary tree architecture Qian et al. (2010)	Schüldet dataset Schuldt et al. (2004)
	Least squares SVM Hsu et al. (2018)	Real data collected from participants and approved by National Cheng Kung University Hospital

acceleration data (Yang et al. 2008). The goal is to recognize more complicated ADL using a tri-axial accelerometer. Ronao et al. also proposed a deep convolutional network to recognize human activities using accelerometers and gyroscope tri-axial sensors. This method outperforms other state-of-the-art data mining techniques, such as SVM for moving activities. However, their experiments are conducted on a small-size dataset (Ronao and Cho 2016). Recently, a deep belief network (DBN) was proposed to recognize human activities using real-world wearable sensor datasets for twelve activities. The limitation is the experimental results were obtained from only one dataset. Besides, the performance results were not compared to other deep learning methods (Hassan et al. 2018). To make a smart environment smarter, certain building parameter such as electricity efficiency, water efficiency, indoor temperature, ventilation rate to name a few must be modelled and improved over time.

Much research has been conducted into human activity recognition using SVM as well. Yin et al. proposed a novel method for detecting abnormal human activities using the one-class SVM (Yin et al. 2008). Correct detection of abnormal human activities yields fewer false positive and can be improved when using accurate abnormal training data. Their method follows a two-step process: firstly, the built a normal activity dataset using the one-class SVM, after which it is passed to a second classifier to reduce the number of false-positives. Qian et al. went further on human activities recognition research by proposing the multi-class SVM (Qian et al. 2010). This method outperforms other state-of-the-art approaches on Schüldet's dataset. Recently, Hsu et al. proposed a least squares SVM that recognize ten human daily activities and eleven sport activities in a wearable inertial sensor network. However, one limitation is the method are evaluated in a laboratory environment (Hsu et al. 2018).

3 Materials and methods

The scope of this research is to recognize human activities using a new supervised algorithm, called MSA. The proposed MSA generates decision boundaries that classify the patterns. This section, we firstly present the methodology used for human activity recognition. Then, presents the MSA training procedure and algorithm in detail.

3.1 Methodology

The proposed methodology for human activity recognition is depicted in Fig. 1. The raw data includes a collection of sensor readings with annotated activities at various timestamps. The annotation indicates the beginning and the end-of-activity occurrence. After raw activity data is gathered, a preprocessing step is performed. This involves transforming the data to the formats of activity label vector that MSA requires. The activity label vector is the data with activity labels. The data is split into training data and test data using a four-fold cross validation. The details of the data preprocessing are discussed in the next section. The activity label vector is the input for the MSA training stage. In this stage, MSA training generates hypersphere prototypes and optimize them through two phases: evolution and partition. The optimized prototypes are the MSA model. The MSA model takes test data as the input to perform classification. After comparison of the predicted activity label and the truth label, we output the performance accuracy.

3.2 MSA training

Statistical learning algorithms predict some future output y based on some data input x, in other words, their goal is to make proper classification of a test dataset as learned from a training set. Given a training data, the learning algorithm

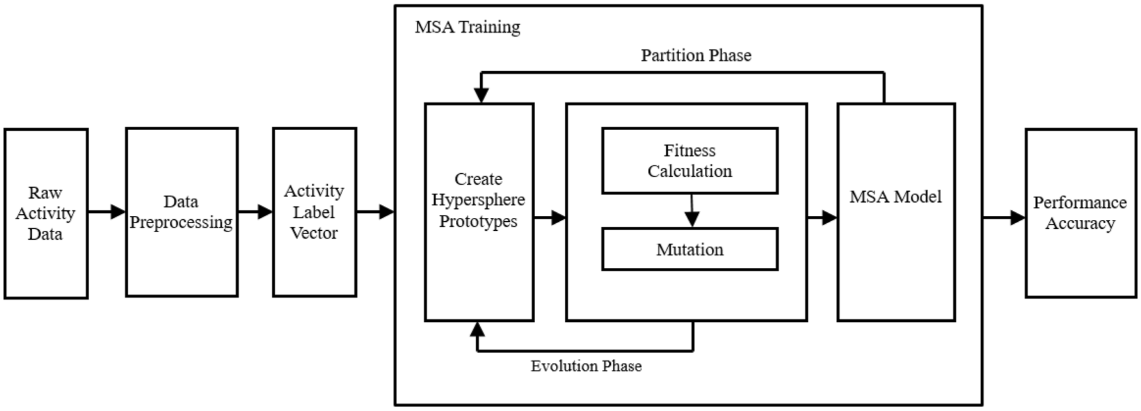

Fig. 1 Human activity recognition using MSA

works by separating data points into different classes. This classification process yields the training model. When a new data is presented, they are fed into the generated training model to test them whether it belongs to a class. This summarizes the mechanism behind most machine learning algorithm when dealing with classification. Similarly, the working principle of MSA is to classify patterns.

MSA is a powerful novel supervised learning algorithm that has been applied for segment analysis in hyperspectral images and noise removal (Fu et al. 2010; Wang et al. 2015a; Wang et al. 2015b; Wang et al. 2016). Later, it is also designed to detect false data injection cyber-physical attack in smart grids (Wang et al. 2017). Recently, Wang et al. carried out a study on the impact of the algorithm parameter called margin on the performance of MSA, by comparing it with SVM. It is concluded that margin is an important parameter that affects all margin-based algorithms (Wang et al. 2018). MSA's decision boundaries are called prototypes. The prototype is a hypersphere with a center, radius and class in a x-dimensional space where $x > 1$. Let G denote the prototype:

$$G = \left(\omega_i, R_i, C_p\right), 1 \le i \le N, p = 1, 2, \dots, P, \tag{1}$$

where ω_i denotes the center of the prototype, R_i is the radius while C_p is the class label. $1 \le n \le X$ is a number set by the user.

From Fig. 1, it can be observed that MSA is implemented as two phases: an evolution phase and partition phase, which are conditioned to execute iteratively and stops after reaching a condition set by the user.

(A) Evolution phase

Evolution phase contains two steps: (1) create hyperspheres prototypes; (2) fitness calculation and mutation. Create initial prototypes by throwing n-random points and calculate the Euclidean distance E_d from the prototype G to each of the subclasses y. For each random data point, find the closet training data sample. Suppose this closest training data sample belongs to class C, find the closet distance from this training data sample to nearest member of a different class C′. This distance is the radius of a prototype.

The constructed initial prototypes contain redundancy. Let us consider Fig. 2 consisting of two classes of sample sets. The 12 red and 12 blue points are the points we want to classify. The constructed prototypes in two dimensional spaces are seven red circles and three blue circles shown in Fig. 2a. The disjunction of the areas that the red circles covers are initial classifiers for red points. The disjunction of the areas that the blue circles covers are initial classifiers for blue points. It is obvious that some red circles are redundant, since some

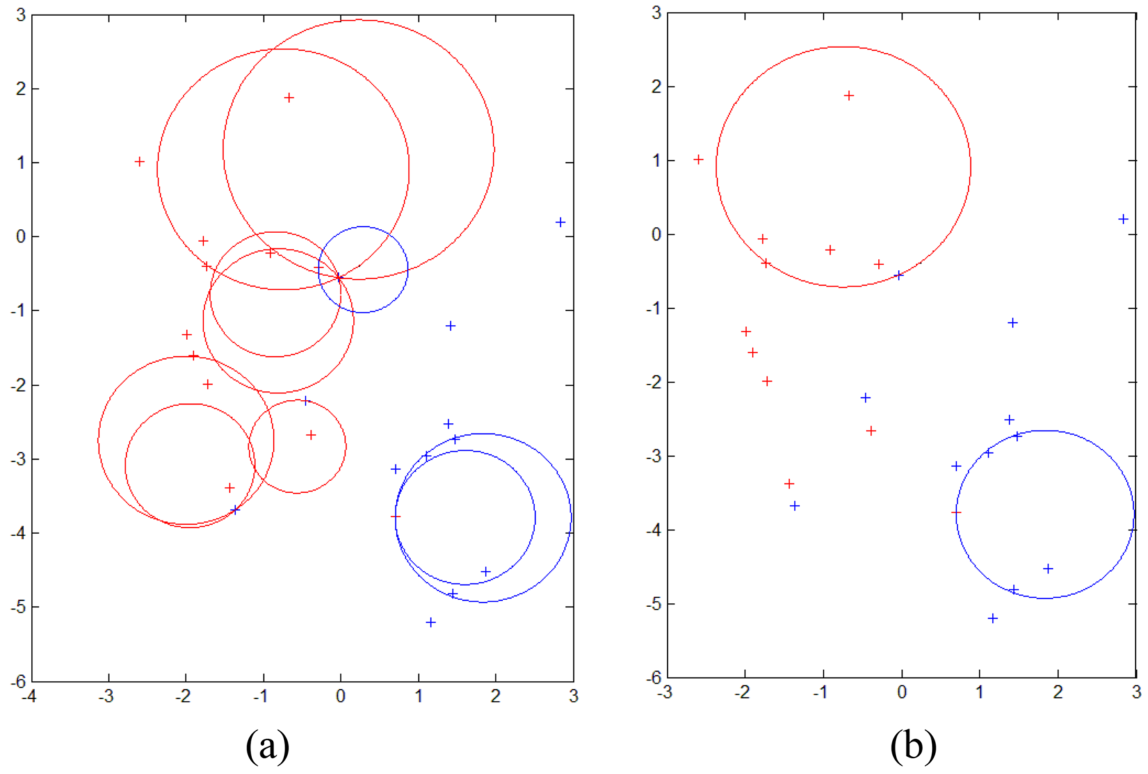

Fig. 2 MSA prototypes construction. **a** Initial prototypes with redundancy; **b** prototypes of the highest FOM

red points can be covered by one big red circle instead of two smaller circles.

Fitness calculation and mutation: To remove the redundancy, MSA employ the next step called fitness calculation and mutation to find the optimal prototypes during each partition phase. The optimal prototypes are the prototypes that has the largest number of sample points inside it. The number of sample points inside the prototype is called the Figure of Merit (FOM). In this case, the optimal prototypes are with highest FOM. Each mutation tends to produce its own FOM. Mutation starts to find the optimal protypes with higher FOM by until we reach a stopping condition is called the within the evolution stage. For each class, doing mutation separately by stochastically select one prototype's centroid to mutate. Selection is based on a probability distribution function proportional to the prototype's figure of merit value. Specifically, it iteratively generates prototypes in the neighborhood areas to find prototypes of highest FOM for each class. The prototypes with highest FOM for each class are shown in Fig. 2b. They cover four red points and four blue points. Both of them has a FOM of 4.

(B) Partition phase

The partitioning phase executes by dividing the training datasets into smaller subsets and trains the hypersphere decision boundaries using an algorithmic parameter: Margin.Definition 1 Margin: it is defined as the percentage χ that the radius of hypersphere shrinks, so radius of χ margin is of the quantity:

$$R_\chi = (1 - \chi)R_0. \tag{2}$$

where $0 \leq \chi < 1$. R_χ is the radius that shrinks χ from R_0. R_0 is called zero margin when $\chi = 0$. It can be seen in Fig. 3, the 12 red and 12 blue points are partially classified by prototypes as red circles and blue circles. The big red circle's margin is 0. After we shrink the circle's radius by 20%, i.e., the smaller red circle has a margin = 0.2. Margin parameter can be adjusted to improve the performance.

The unique feature of MSA enables it to easily solve non-separable cases and datasets with lots of features. Let D denote a dataset, with subsets partitioned by MSA as $D_1, D_2, D_3 \ldots.$, where they can be viewed as a union of D.

$$D = \cup_{j>1}^{D} D_j \quad D_j \neq \emptyset. \tag{3}$$

After optimal prototypes are found, the sample points inside the prototypes are removed from the sample dataset. Then we throw randomly points among the remaining sample points and start construct initial prototype again. The process continues until no sample points are left. When presented with a test dataset, it is likely that some misclassified or unclassified datapoints will be encountered. When this happens, we can opt to set the number of generations as a parameter to measure the number of partition phase executed. After it reaches to the number of generations, the algorithm is terminated.

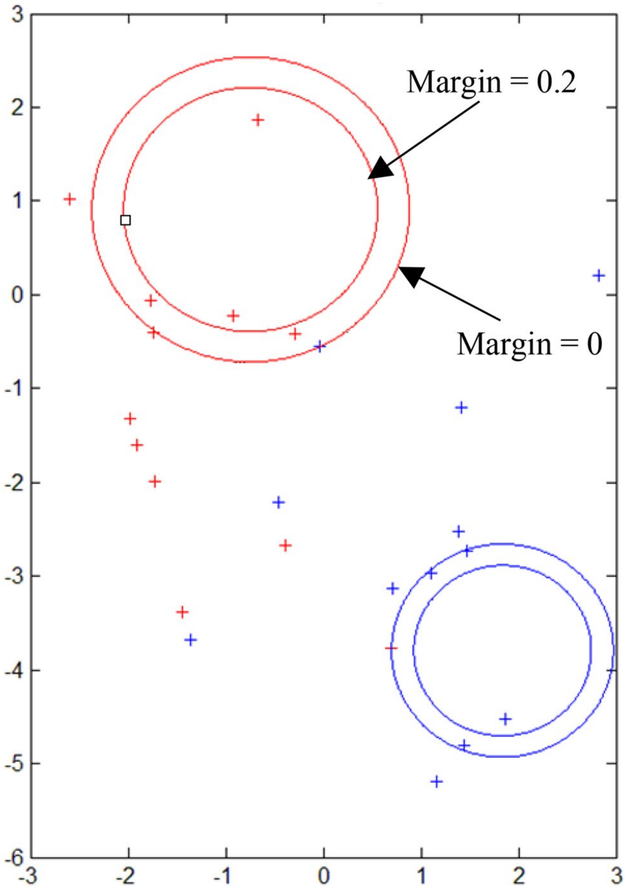

Fig. 3 Margin in MSA

The performance of MSA can be tuned when an appropriate margin is selected. Smaller margin has less data unclassified, more false positives with less iterations while larger margin has more data unclassified, fewer false positives with more iterations. This tradeoff in choosing an appropriate margin size for a better performance accuracy prompted the work on the next section.

3.3 Algorithm

We will summarize the step by step approach of MSA in this section. This will give a better understanding of the said algorithm. MSA consists of two phases: partitioning and evolution. Four parameters should be considered.

Step 1. Initialization

Maximum Mutation (MM), $MM = 20$;

Maximum Generation (MG), $MG = 20$;

Number of random points (N), $N = 20$;

χ: margin, $\chi = 0$;M : number of iterations during mutation, M = 0;Q : number of partitions/generations, Q = 0;

Step 2. Training presentation

Given a multiclass P(P > 1) classification problem, training set T has m sample points, denoted by $T = \left\{\left(x_1, \dots, x_m\right)\right\}$. Each sample point $x_k (1 \leq k \leq m)$ is n-dimension $(n \geq 2)$ vector. Each training data sample x_k has class labeled $C_p (p = 1, \dots, P)$. Normalize T into [0, 1] space. Generate N random points that are uniform distributed in [0, 1] space, denoted as $\mathrm{R} = \left\{\left(\omega_1, \dots, \omega_\mathrm{N}\right)\right\} \in$ Unif [0, 1].

Step 3. Create prototypes

Build initial prototypes $G_i = \left(\omega_i, R_i, C_p\right), 1 \leq i \leq N, p = 1, 2, \dots, P\}$. Each random point $\omega_i (1 \leq i \leq N)$ is n-dimensional $(n \geq 2)$ vector. ω_i will serve as the centroid of G. For each class C_i, find the ω_i that is the minimum Euclidean distance from ω_i to x_k:

$$\min \left\|\omega_i - x_k\right\|. \tag{4}$$

The radius of prototypes R_k equals to the following minimum Euclidean distance from ω_i to x_j:

$$R_k = \min \left\|\omega_i - x_j\right\|, C_j \neq C_k. \tag{5}$$

Step 4. Fitness calculation

Compute the fitness, measured by figure of merit of the prototype. The figure of merit of prototype G_i is denoted as $\mathrm{F}_{\mathrm{G}_\mathrm{i}}$, i.e., the number of class C_p data samples inside of G_i geometrically. Suppose class label C_p contains total h prototypes for during the current iteration. The largest figure of merit among all h prototypes is LF:

$$LF = \max\left\{\mathrm{F}_{\mathrm{G}_1}, \mathrm{F}_{\mathrm{G}_2}, \dots, \mathrm{F}_{\mathrm{G}_\mathrm{h}}\right\}. \tag{6}$$

Step 5: Mutation

Stochastically select a center ω_i' of prototype G_i to mutate. ω_i' is selected based on a probability distribution function proportional to the prototype's figure of merit value. Calculate the proportional of the prototypes of figure of merit f_p:

$$f_p = \frac{F_{G_i}}{\sum_1^h F_{G_i}}. \tag{7}$$

If i in ω_i' satisfy the following probability distribution function and $\zeta \in$ Unif [0, 1]:

$$\sum_{\xi=1}^{i-1} f_\xi < \zeta \leq \sum_{\xi=1}^{i} f_\xi. \tag{8}$$

Select ω_i' to mutate to another N points. The mutated N points are:

$$\omega_i' + \varepsilon \alpha U, \tag{9}$$

where ε is random sign symbol $\{-1,1\}$. $\alpha \in$ Unif [0, 1]. U is the maximum perturbation:

$$U = \begin{cases} \omega_i' & if \omega_k \leq \frac{\min\{x_k\}+\max\{x_k\}}{2} \\ \max\left\{x_k\right\} - \omega_i' & Otherwise \end{cases} (1 \leq k \leq m). \tag{10}$$

The number of iterations during mutation M increment by 1 to M + 1. If $M \geq MW$ or $\mathrm{LF}^M > \mathrm{LF}^{\mathrm{M}+1}$ (compares LF between two iterations M and M + 1), go to the next step 6. Otherwise, repeat the process by going back to Step 3. The mutated N points are random points ω_i to create prototypes again.

Step 6. Partition

Partition the training set and yield the optimum prototypes G_i^o. The number of generations Q increment by 1 to Q + 1. Store the prototype G_i^o with largest figure of merit LF^M, and radius $R_{i,Q}$. Remove all data samples falling inside of G_i^o of the current generation Q. The radius of G_i^Q is $R_{i,Q}$:

$$R_{i,Q} = (1 - \chi) R_{i,Q}. \tag{11}$$

We set the margin χ to shrink the radius of hyperspheres. Store the reduced training set T′. If $\mathrm{T}' \neq \emptyset$ and Q < MQ, repeat the process by going back to Step 2. Training set T is updated as T′.

Step 7. Classification

If there are P classes in the training set T, the prototypes, i.e., the decision boundaries of MSA is the union of prototypes for all P classes generated in total Q generations:

$$G'' = \bigcup_{i=1}^{Q} \bigcup_{j=1}^{P} G_i. \tag{12}$$

For all points y_i in test set S, the Euclidean distance $\left|\left|y_i - \omega_k\right|\right|$ is computed between y_i and ω_k, where ω_k is the center of prototypes G'':

$$\left\|y_i - \omega_k\right\| \leqslant R_k. \tag{13}$$

If the above inequality holds, y_i is with class label C_p, where $(\omega_k, R_k, C_p) \int G''$.

4 Experiments and result

4.1 Datasets

To evaluate the effectiveness of the proposed MSA algorithm for human activity recognition, we select two real-world data sets in the smart home environment.

(1) ARAS dataset (Alemdar et al. 2013). This dataset contains two houses A and B, with four residents, two in each of the houses, of which 27 different human activities were recorded which was performed for 30 days on 20 different sensors, as shown in Table 2. The sensor ID, type and location are demonstrated as the ambient sensor descriptions. Sensor ID is a unique identifier for the sensor. Among these sensors, they are classified as seven types. They are photocell to detect open drawers and wardrobes activities, which are located in the drawers, the wardrobes and the refrigerator. IR (infrared) sensor is located near the TV to detect watch TV activity. Force sensor or Pressure Mat are sensors under the bed and the couches to detect sleeping, sitting, and napping actions. Contact sensors are put in the door frames, shower cabins and cupboard to detect actions of opening and closing of the doors and cupboards. Sonar distance sensor is detecting presence, which are located on the walls and door frames. Temperature sensors recognize cooking activity which are near the oven in the kitchen. The size of House A is 538 square feet. It includes one bedroom, one living room, one kitchen and one bathroom. The two residents in House A are two males both aged 25. They are denoted as resident 1 and 2. The size of House B is 968 square feet. It includes 2 bedrooms, one living room, on kitchen and one bedroom. The two residents in House B is a married couple both aged 34. They are denoted as resident 3 and 4. There are 27 human daily activities in both House A and House B. Activities include: other, going out, preparing breakfast, having breakfast, preparing lunch, having lunch, preparing dinner, having dinner, washing dishes, having snack, sleeping, watching TV, studying, having shower, toileting, napping, using internet, reading book, laundry, shaving, brushing teeth, talking on the phone, listening to music, cleaning, having conversation, having guest, and changing clothes. Sensor reading was recorded per second with a total of 86,400 data points for each day with the timestamp from 00:00:00 to 23:59:59. One example of the house A activities is shown in the left part of Table 3.

(2) CASAS dataset (Cook et al. 2012). This dataset was collected from Tulum test bed from the Washington State University CASAS smart home project (Cook and Schmitter-Edgecombe 2009). This smart home includes three bedrooms, one bathroom, one kitchen and a living/dining room. The normal daily activities were

Table 2 Ambient sensor descriptions of ARAS dataset

Column	House A			House B		
	Sensor ID	Sensor type	Location	Sensor ID	Sensor type	Location
1	Ph1	Photocell	Wardrobe	Co1	Contact sensor	Kitchen cupboard
2	Ph2	Photocell	Convertible	Co2	Contact sensor	Kitchen cupboard
3	Ir1	IR	TV receiver	Co3	Contact sensor	House door
4	Fo1	Force sensor	Couch	Co4	Contact sensor	Wardrobe door
5	Fo2	Force sensor	Couch	Co5	Contact sensor	Wardrobe door
6	Di3	Distance	Chair	Co6	Contact sensor	Shower cabinet
7	Di4	Distance	Chair	Di2	Distance	Tap
8	Ph3	Photocell	Fridge	Fo1	Force sensor	Chair
9	Ph4	Photocell	Kitchen drawer	Fo2	Force sensor	Chair
10	Ph5	Photocell	Wardrobe	Fo3	Force sensor	Chair
11	Ph6	Photocell	Bathroom cabinet	Ph1	Photocell	Fridge
12	Co1	Contact sensor	House door	Ph2	Force sensor	Kitchen drawer
13	Co2	Contact sensor	Bathroom door	Pr1	Pressure mat	Couch
14	Co3	Contact sensor	Shower cabinet	Pr2	Pressure mat	Couch
15	So1	Sonar distance	Hall	Pr3	Pressure mat	Bed
16	So2	Sonar distance	Kitchen	Pr4	Pressure mat	Bed
17	Di1	Distance	Tap	Pr5	Pressure mat	Arm chair
18	Di2	Distance	Water closet	So1	Sonar distance	Bathroom door
19	Te1	Temperature	Kitchen	So2	Sonar distance	Kitchen
20	Fo3	Force sensor	Bed	So3	Sonar distance	Closet

Table 3 Activity descriptions in the datasets

ARAS dataset		CASA dataset	
Activity	Number of events	Activity	Number of events
Other	238	Cook breakfast	80
Going out	683	Cook lunch	71
Preparing breakfast	10	Enter home	73
Having breakfast	21	Group meeting	11
Sleeping	559	Leave home	75
Watching TV	207	R1 eat breakfast	66
Studying	52	R1 snack	491
Toileting	41	R2 eat breakfast	47
Using internet	145	Wash dishes	71
Laundry	22	Watch TV	528
Brushing teeth	10		
Talking on the phone	62		
Changing clothes	21		

recorded from April to July of 2009. Total 18 motion sensors and two temperature sensors are equipped to collect 11 human activities for two married residents. As it can be seen from the right part of Table 3, they are cook breakfast, cook lunch, enter home, group meeting, leave home, eat breakfast, having snack, wash dishes, watch TV, and other activities.

When selecting locations for the sensors in human activity recognition, sensor and activity matching is considered. Human activities can be identified by one or several sensors. It is expected that several actions occur concurrently or successively during the period of the recognized activity. For example, when people are preparing the breakfast (one activity in Table 3), the actions may include opening the kitchen drawer to get the spoon, knife, bowl, opening the fridge to get the food, etc. In this case, sensor readings should be active at least one of the four sensors: Ph3, Ph4, So2 or Te1. As shown in Table 2, Ph3 is photocell sensor in the Fridge, Ph4 is the photocell sensor in the kitchen drawer, So2 is the sonar distance sensor in Kitchen, and Te1 is the temperature sensor in the kitchen. On Day 5 of resident 1 in House A, preparing breakfast activity is labeled in four scenarios: (1) active reading on Tel1; (2) active readings on both So2 and Tel1; (3) active readings on both Ph3 and Tel1; (4) active readings on Ph3, So2 and Tel1.

4.2 Data preprocessing

The data preprocessing part of the experiment includes segmentation, feature extraction, and label combinations. Segmentation is a step that aims to divide the time series raw streaming data that is suitable for activity recognition. A most commonly used method is time-based windowing method (Espinilla et al. 2018). Each sensor event is with the form of date, time, sensor ID, sensor status and activity. Time-based method is implemented to segment the sensor event datasets based on a fixed time slice. Although there are some limitation of this method for choosing a window length, we empirically select the window size. This window size is affected by the average number of activities for making labelling decision. During each time slice, we label the activity that dominates it.

Feature extraction transforms the sensor event after segmentation into a feature vector. Each feature vector consists of sensor signals as their feature values. Sensor signals are discrete binary values, either 0 or 1 for each second. For example, there are 20 sensor signals in ARAS dataset. The dimension of the feature vector is 20. Each feature vector is tagged with the label from two residents.

Label combination aims to deal with multi-label vectors. Since each feature vector contains sensor signals as their features plus two activity labels for two residents, it is a multi-label vector. We employ a label combination approach by transforming a multi-label vector to a single-label vector (Read et al. 2015; Read et al. 2008). Treating each instance data label set as an atomic label, the multi-label $\{o, p, q\}$ can be represented as a single label opq. Therefore, the set of two label datasets is converted to be considered by single-label classifier.

5 Experimental results

In the experiments, we evaluate the effectiveness of the proposed MSA on two benchmark datasets, with comparison to other two popular algorithms: SVM and ANN. The concept behind MSA is the application of a hypersphere to classify correctly class samples, whereas SVM uses a hyperplane and ANN uses perceptron to classify points on a plane. In this study, we reported results for 10 days of activities. The default parameters for all three algorithms are chosen. Number of random points at initialization (NA) = 20. Number of maximum mutation for every generation (NmuMax) = 20. The selected number of maximum generations (NGenMax) = 20. The margin parameter χ is set as zero margin ($\chi = 0$) as default value. The SVM default parameters we chose are Radius basis function (RBF) kernel, with regularization parameter C = 1. The kernel parameter γ is set as the reciprocal of number of features in the data sets. A feed-forward backpropagation network is implemented. It is a two-layer ANN with 10 neurons in its hidden layers. All the rest of the parameters are default values. The result is analyzed and shown on Table 4.

The performance analysis depicts the effectiveness of MSA, compared to SVM and ANN. It is obvious shown

Table 4 Activity recognition results in accuracy (%) between MSA, SVM and ANN

Days	ARAS dataset						CASAS dataset		
	House A			House B			Tulum		
	MSA	SVM	ANN	MSA	SVM	ANN	MSA	SVM	ANN
1	60.32	58.51	59.97	88.56	85.64	86.25	72.24	70.13	70.96
2	76.29	75.14	75.68	91.25	90.21	90.89	70.54	69.52	70.21
3	65.55	63.89	64.44	98.71	97.86	97.22	57.74	56.23	57.35
4	66.85	63.25	63.85	96.84	95.23	96.14	54.15	53.19	53.88
5	69.75	66.56	67.23	97.36	95.15	96.98	75.90	74.21	75.05
6	63.24	61.59	62.84	99.55	98.46	99.21	68.12	66.93	67.55
7	69.84	68.98	68.86	99.28	98.68	98.91	71.32	69.38	70.11
8	76.57	73.24	72.15	95.84	93.87	94.68	67.45	65.25	66.45
9	61.55	59.87	60.22	98.21	97.55	98.08	67.22	66.24	66.95
10	78.59	77.98	78.01	96.83	95.46	95.89	75.36	74.95	75.21
Average	68.85	66.90	67.32	96.24	94.81	95.42	68.00	66.60	67.37

in Table 4 and Fig. 4a, MSA outperforms SVM and ANN for 10 days in house A. In other words, MSA has the lowest number of mis-detected human activities. Specifically, MSA performs much better than SVM in day 4 and day 8, with a significant increase of 3.6% and 3.33% performance accuracy. On day 4, there are 18 daily activities recognized for resident 1 from midnight 0:00:00 to 23:59:59. The sequence of the activities are: watching TV, using the internet, talking on the phone, brushing teeth, sleeping, toileting, other, listening to music, going out, having breakfast, washing dishes, studying, preparing lunch, having lunch, having conversation, having snack, preparing dinner, having dinner. There are 12 daily activities recognized for resident 2 since midnight: using internet, brushing teeth, sleeping, other, toileting, shaving, having shower, preparing breakfast, having breakfast, changing clothes, going out and having conversation. Compare to ANN, MSA's superiority in performance can be also seen on day 8 where its performance accuracy is 76.57% as compared to ANN which is 72.15%. On that day, 18 activities recognized for resident 1 is watching TV, talking on the phone, using the internet, reading book, brushing teeth, listening to music, having conversation, sleeping, toileting, other, changing clothes, going out, having shower, studying, having snack, preparing dinner, having dinner and washing dishes. Meanwhile, 18 activities are recognized from resident 2. Compare to resident 1, there are two different activities: napping and laundry. Besides, MSA yields 2.52% performance gain on day 5, comparing to ANN. The average performance accuracy of MSA is 1.95% better than SVM and 1.53% better than ANN.

Compare to the daily activities of two males residents in House A, the detection accuracy is generally higher in

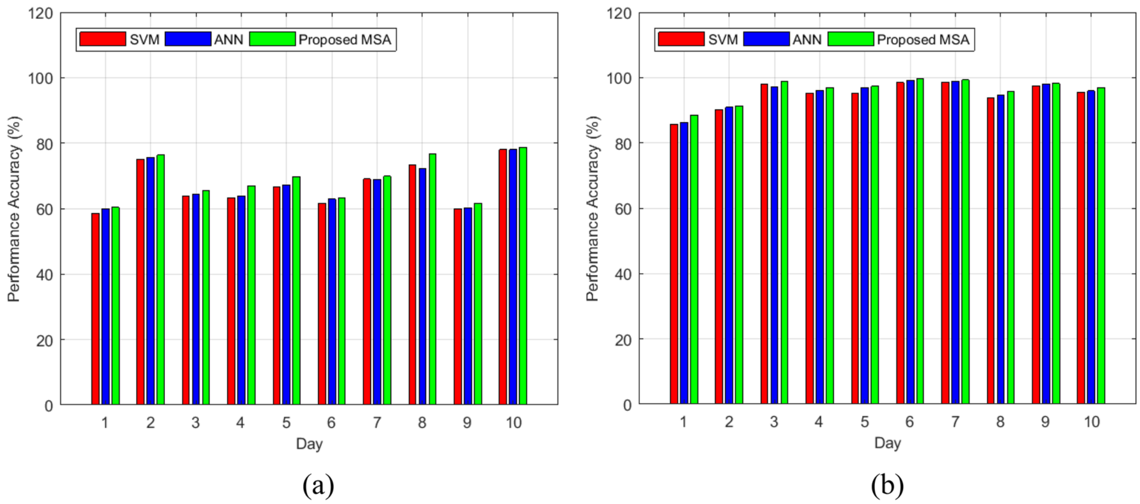

Fig. 4 Activity recognition performance accuracy of the proposed MSA, SVM and ANN on ARAS dataset: **a** house A, **b** house B

House B for the married couple in House B. One reason is that the couple have less, but more regular daily activities than two male residents in House A. The depiction of the performance of Table 4 and Fig. 4b of house B yield much better accuracy in comparison to house A. On day 1, the performance accuracy of MSA is 88.56%, compared to SVM and ANN with 85.64% and 86.52%, respectively. MSA shows an performance increase of 2.92%. Another day 5, MSA yields a recognition accuracy of 97.36%, whereas SVM can only reaches the performance of 95.15%, which is 2.21% performance loss. On day 5, the recognized activities from midnight for the couples are only 8 activities (sleeping, toileting, having shower, changing clothes, other, going out, having conversation, watching TV) for resident 3, and 9 activities (sleeping, other, using internet, napping, toileting, changing clothes, going out, having conversation, watching TV) for resident 4. There was no scenario that SVM or ANN had a better performance accuracy than MSA. That means that both SVM and ANN has a greater number of wrongly detected human activities than MSA. When the total performance accuracy of the three algorithms are averaged, MSA outputs 1.43% and 0.82% better than SVM and ANN.

Table 4 also reports the activity recognition performance on CASAS dataset. MSA again performs better than SVM and ANN in all 10 days' experiment. The average performance of MSA is 68%, comparing to 66.60% of SVM and 67.37% of ANN. During some days, such as day 1, 7 and 8, recognition performance has been improved significantly, with 2.11%, 1.94% and 2.2% increase when using MSA, comparing to SVM. For example, MSA also outputs a much better performance than ANN in day 1. This day was recorded on April 2, 2009. The recognized activities are cooking breakfast, eating breakfast, cooking lunch, leaving home, watch TV, eating snack, entering home. MSA outputs an accuracy of 72.24%, whereas ANN outputs an accuracy of 70.96%. Overall, in all set of experiments, the classification performance of MSA is close or better than the ones obtained by SVM and ANN.

5.1 Computational complexity of MSA

The experiments are carried out using a computer with 3.4 GHz processor and 8 GB RAM. All three algorithms: MSA, SVM and ANN are implemented in MATLAB environment. However, the SVM classification used the LibSVM programming library (Chang and Lin 2011). The library provides MEX file that calls the SVM implementation written in C language. Therefore, classification used by SVM is much faster than MSA and ANN. MSA shows a comparable time performance as ANN. The running time using 1 day event of ARAS data set is reported in Table 5 as follows.

Table 5 Running time of MSA, SVM and ANN

	Training time (s)	Test time (s)
MSA	7.25	2.63
ANN	7.31	2.35
SVM	2.16	1.31

It can be seen that both the training and test time in SVM is very fast due to the implementation in C language. MSA yields similar performance as ANN with the implementation in MATLAB language. However, it is expected that MSA yield better performance than SVM if MATLAB built-in SVM function is used. This comes from the fact that MSA does not project data to the high-dimensional kernel space. This kernel mapping adds more computational complexity to SVM.

To further investigate the time complexity of the algorithm, we employed the Big-O notation analysis on MSA. We calculate the running time in the implementation code and derive a running time function. N is denoted as the amount of training data as the input to MSA. They asymptotic complexity of MSA during training phase is presented as follows:

$$T(N) = O\left(Q \times \left(N^2 + M * N^2\right)\right) = O\left(N^2\right), \tag{14}$$

where M is the number of iterations during mutation. Q is the number of generations. Since both M and Q are constant numbers which does not grow with N, the training time of MSA is bounded by a quadratic polynomial $O\left(N^2\right)$.

6 Conclusion

This research proposes to apply a supervised machine learning algorithm called MSA for human activity recognition. This research takes it further by applying it to activity recognition datasets to understand its effectiveness while using the smart home as a case study. The results were promising when applied to ARAS dataset for two houses A and B and CASAS dataset. The experimental results demonstrate the MSA yields close performance or outperforms SVM and ANN. However, the proposed approach has several limitations in human activity learning. First, this approach does not yield comparable performance with deep learning algorithms. We conduct an experiment to compare the proposed approach with convolutional neural network (CNN). The activity datasets of two male residents in House A of CASAS is used. The results show that the CNN have much better classification accuracy performance than MSA in Table 6. In the future, we will improve the classification performance of MSA using kernel method. This method can map the data from the original feature space to a higher dimensional feature space to classify data through hypersphere boundaries.

Table 6 Compare performance of MSA with CNN

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
MSA	60.32	76.29	65.55	66.85	69.75	63.24	69.84	76.57	61.55	78.59
CNN	65.34	80.41	69.47	70.48	73.57	67.12	73.30	80.79	65.47	81.87

The motivation is to classify non-linear patterns efficiently in high-dimensional spaces to obtain an adequate representation power of MSA.

Second, the impact of parameters of MSA is not fully considered in this experiment. To be consistent with other algorithms, default values of the parameters are chosen. The performance of MSA can be improved if we tune the parameters of the algorithms. Third, the data collected comes are small-scale human activity datasets. One future research is to optimize the implementation of MSA using parallelism to process large datasets for activity recognition (Wang et al. 2019). On a final note, we believe this research has contributed tremendously to the field of activity recognition using a novel supervised learning approach. To the best of our knowledge, this is the first research to apply MSA to activity recognition in a smart home environment.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alemdar H, Ertan H, Incel OD, Ersoy C (2013) ARAS human activity datasets in multiple homes with multiple residents. In: Proceedings of the 7th international conference on pervasive computing technologies for healthcare, 2013. ICST (Institute for Computer Sciences, Social-Informatics and Technology), pp 232–235

Batchuluun G, Kim JH, Hong HG, Kang JK, Park KR (2017) Fuzzy system based human behavior recognition by combining behavior prediction and recognition. Expert Syst Appl 81:108–133

Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: Tenth IEEE international conference on computer vision (ICCV'05) Volume 1, vol. 2, pp 1395–1402. IEEE

Brendel W, Fern A, Todorovic S (2011) Probabilistic event logic for interval-based event recognition. In: CVPR 2011, 2011. IEEE, pp 3329–3336

Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2:1–27

Chen L, Nugent CD (2019) Sensor-based activity recognition review. Human activity recognition and behaviour analysis. Springer, New York, pp 23–47

Chen Y-P, Yang J-Y, Liou S-N, Lee G-Y, Wang J-S (2008) Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. Appl Math Comput 205:849–860

Cook DJ, Schmitter-Edgecombe M (2009) Assessing the quality of activities in a smart environment. Methods Inf Med 48:480–485

Cook DJ, Crandall AS, Thomas BL, Krishnan NC (2012) CASAS: a smart home in a box. Computer 46:62–69

Cook DJ, Krishnan NC, Rashidi P (2013) Activity discovery and activity recognition: a new partnership. IEEE Trans Cybern 43:820–828

Cornacchia M, Ozcan K, Zheng Y, Velipasalar S (2016) A survey on activity detection and classification using wearable sensors. IEEE Sens J 17:386–403

Dahmen J, Thomas BL, Cook DJ, Wang X (2017) Activity learning as a foundation for security monitoring in smart homes. Sensors 17:737

Emphasis Telematics, https://www.emphasisnet.gr/e-glove/. Accessed Nov 2019

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 Hum Comput 1–13

Fu J, Caulfield HJ, Wu D, Tadesse W (2010) Hyperspectral image analysis using artificial color. J Appl Remote Sens 4:043514

Hassan MM, Huda S, Uddin MZ, Almogren A, Alrubaian M (2018) Human activity recognition from body sensor data using deep learning. J Med Syst 42:99

Holzinger A, Röcker C, Ziefle M (2015) From smart health to smart hospitals. Smart health. Springer, New York, pp 1–20

Hsu Y-L, Yang S-C, Chang H-C, Lai H-C (2018) Human daily and sport activity recognition using a wearable inertial sensor network. IEEE Access 6:31715–31728

Igwe OM, Wang Y, Giakos GC (2018) Activity learning and recognition using margin setting algorithm in smart homes. In: 2018 IEEE ubiquitous computing, electronics and mobile communication conference (UEMCON), New York, Nov 8–10, 2018. IEEE, pp 294–299

Lillo I, Soto A, Carlos Niebles J (2014) Discriminative hierarchical modeling of spatio-temporally composable human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014. pp 812–819

Liouane Z, Lemlouma T, Roose P, Weis F, Messaoud H (2018) An improved extreme learning machine model for the prediction of human scenarios in smart homes. Appl Intell 48:2017–2030

Liu L, Wang S, Su G, Huang Z-G, Liu M (2017) Towards complex activity recognition using a Bayesian network-based probabilistic generative framework. Pattern Recogn 68:295–309

Malaisé A, Maurice P, Colas F, Charpillet F, Ivaldi S (2018) Activity recognition with multiple wearable sensors for industrial applications. In: ACHI 2018-eleventh international conference on advances in computer–human interactions, 2018

Medina-Quero J, Zhang S, Nugent C, Espinilla M (2018) Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition. Expert Syst Appl 114:441–453

Mohamed R, Perumal T, Sulaiman MN, Mustapha N (2017) Multi resident complex activity recognition in smart home: a literature review. Int J Smart Home 11:21–32

Ordóñez FJ, Roggen D (2016) Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 16:115

Palumbo F, Gallicchio C, Pucci R, Micheli A (2016) Human activity recognition using multisensor data fusion based on reservoir computing. J Ambient Intell Smart Environ 8:87–107

Qian H, Mao Y, Xiang W, Wang Z (2010) Recognition of human activities using SVM multi-class classifier. Pattern Recogn Lett 31:100–111

Read J, Pfahringer B, Holmes G (2008) Multi-label classification using ensembles of pruned sets. In: 8th IEEE international conference on data mining, 2008. IEEE, pp 995–1000

Read J, Martino L, Olmos PM, Luengo D (2015) Scalable multi-output label prediction: from classifier chains to classifier trellises. Pattern Recognit 48:2096–2109

Reyes-Ortiz J-L, Oneto L, Samà A, Parra X, Anguita D (2016) Transition-aware human activity recognition using smartphones. Neurocomputing 171:754–767

Roetenberg D, Luinge H, Slycke P (2009) Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors Xsens Motion Technologies BV, Tech Rep 1

Roggen D et al (2010) Collecting complex activity datasets in highly rich networked sensor environments. In: 2010 Seventh international conference on networked sensing systems (INSS), 2010. IEEE, pp 233–240

Ronao CA, Cho SB (2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244

Saha HN, Mandal A, Sinha A (2017) Recent trends in the Internet of Things. In: 2017 IEEE 7th annual computing and communication workshop and conference (CCWC), 2017. IEEE, pp 1–4

Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004, 2004. IEEE, pp 32–36

Sezer OB, Dogdu E, Ozbayoglu AM (2017) Context-aware computing, learning, and big data in internet of things: a survey. IEEE Internet Things J 5:1–27

Van Laerhoven K, Aidoo KA, Lowette S (2001) Real-time analysis of data from many sensors with neural networks. In: Proceedings fifth international symposium on wearable computers, 2001. IEEE, pp 115–122

Wang Y, Adhami R, Fu J (2015) A new machine learning algorithm for removal of salt and pepper noise. In: Seventh international conference on digital image processing (ICDIP 2015), 2015a. International society for optics and photonics, p 96311R

Wang Y, Adhmai R, Fu J, Al-Ghaib H (2015b) A novel supervised learning algorithm for salt-and-pepper noise detection. Int J Mach Learn Cybern 6:687–697

Wang Y, Fu J, Adhami R, Dihn H (2016) A novel learning-based switching median filter for suppression of impulse noise in highly corrupted colour images. Imaging Sci J 64:15–25

Wang Y, Amin M, Fu J, Moussa H (2017) A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids. IEEE Access 5:26022

Wang Y, Fu J, David Pan W (2018) Impact of setting margin on margin setting algorithm and support vector machine. J Imaging Sci Technol 62:30501–30511

Wang Y, Fu J, Wei B (2019) A novel parallel learning algorithm for pattern classification SN. Appl Sci 1:1647

Yang J-Y, Wang J-S, Chen Y-P (2008) Using acceleration measurements for activity recognition: an effective learning algorithm for constructing neural classifiers. Pattern Recognit Lett 29:2213–2220

Yin J, Yang Q, Pan JJ (2008) Sensor-based abnormal human-activity detection. IEEE Trans Knowl Data Eng 20:1082–1090

Zhang S, Wei Z, Nie J, Huang L, Wang S, Li Z (2017) A review on human activity recognition using vision-based method. J Healthc Eng 2017:1–31. https://doi.org/10.1155/2017/3090343