Genetic algorithm–optimized support vector machine for real-time activity recognition in health smart home

Yan Hu1,2, Bingce Wang1, Yuyan Sun3,4, Jing An5 and Zhiliang Wang1

Abstract
Health smart home, as a typical application of Internet of things, provides a new solution for remote medical treatment. It can effectively relieve pressure from shortage of medical resources caused by aging population and help elderly people live at home more independently and safely. Activity recognition is the core of health smart home. This technology aims to recognize the activity patterns of users from a series of observations on the user’s actions and the environmental conditions, so as to avoid distress situations as much as possible. However, most of the existing researches focus on offline activity recognition, but not good at online real-time activity recognition. Besides, the feature representation techniques used for offline activity recognition are generally not suitable for online scenarios. In this article, the authors propose a real-time online activity recognition approach based on the genetic algorithm–optimized support vector machine classifier. In order to support online real-time activity recognition, a new sliding window-based feature representation technique enhanced by mutual information between sensors is devised. In addition, the genetic algorithm is used to automatically select optimal hyperparameters for the support vector machine model, thereby reducing the recognition inaccuracy caused by manual tuning of hyperparameters. Finally, a series of comprehensive experiments are conducted on freely available data sets to validate the effectiveness of the proposed approach.

Keywords
Health smart home, activity recognition, support vector machines, genetic algorithm, mutual information

Date received: 19 August 2020; accepted: 6 October 2020

Handling Editor: Francesc Pozo

Introduction
Nowadays, improvements in medicine have increased the average age of the world’s population. The United Nations predicts that by 2050, 22% of the world’s population will be above 65 years of age. As a result, most nations have to face the demographic modification problem and need to develop a series of healthcare technologies to help elderly people live their life in good conditions. In modern society, the avoidable distance between family members makes elderly people often live alone and has to be autonomous. Moreover, with the increase in life expectancy, diseases such as Alzheimer’s become more and more prevalent. To avoid distress
situations (fall for instance) as much as possible, tele-monitoring technologies should be developed to detect significant changes in the activities or habits of a person and help elderly people stay at home independently and safely.

Health smart homes equipped with various sensors provide an excellent solution for telemonitoring. The MIT (Cambridge, MA, USA) project House_n is a good example. In this project, hundreds of sensors are installed in a flat to monitor users’ daily activities. In addition, users are provided with a series of human-machine interfaces to control their environments, which can help people maintain physical and mental health. The Georgia Institute of Technology works on “The Aware Home Research Initiative,” a two-floor smart home designed to consider the different requirements of children with mental disabilities and the elderly people of a family. Researchers utilize motion and environmental sensors, video cameras and Radio Frequency IDentification (RFID) tags deployed in the house to explore ways to help people live independently and safely at home when they are old or handicapped. In France, the researchers of both AILISA and PROSAFE projects use presence infrared sensors to monitor users’ activities and raise alarms in case of anomalies.

A large number of sensors deployed in a smart home environment definitely produce a large volume of heterogeneous and multidimensional streaming data. Therefore, developing effective data processing technologies to detect anomalies inside the large amount of information is a big challenge. This article presents our research and contribution to the automatic activity recognition technology in health smart homes. We propose a novel activity recognition approach based on a multi-class support vector machine (SVM) framework. A sliding window–based feature representation method enhanced by sensor mutual information is designed to support online real-time activity recognition. In addition, the SVM classification model is optimized by the genetic algorithm to automatically select the optimal hyperparameters. This approach can effectively identify multiple categories of daily activities of the elderly people. The key contributions of this work are summarized as follows:

- We design a sliding window–based feature extraction method, which can effectively reduce the influence of irrelevant information contained in a time window of sensor events by incorporating sensor mutual information into the feature vector, thereby improving the accuracy of activity recognition.
- We propose a multi-class SVM classification framework based on the above feature extraction technique to realize online real-time activity recognition. The genetic algorithm is employed to automatically select optimal hyperparameters for the SVM classifier, thereby reducing the recognition inaccuracy caused by overdependence on human experience.
- Comprehensive experiments are conducted over freely available data sets to validate the effectiveness of the proposed SVM-based activity recognition approach.

The rest of the article is organized as follows. The research literature about activity recognition is presented in section “Related work.” Section “Preliminaries” introduces the SVM theory, and then presents the problem statement and the approach overview. In section “Activity recognition based on genetic algorithm–optimized SVM classifier,” we elaborate on the proposed activity recognition technology, including segmentation of sensor event sequence, feature extraction, and genetic algorithm–optimized SVM for activity recognition. Comprehensive experiments are conducted in section “Experiments and discussion” to verify the effectiveness of the proposed approach. Finally, we draw a conclusion in section “Conclusion.”

**Related work**

**Activity monitoring technology**

Monitoring user actions and environmental changes is fundamental to human activity recognition. According to different monitoring equipments, activity monitoring technology is generally divided into two categories: video-based and sensor-based.

**Video-based activity monitoring.** Video-based activity monitoring continuously tracks and records user actions through cameras installed in a smart environment. Then, a series of two- or three-dimensional (2D or 3D) images are processed by specific algorithms for activity recognition, where 2D images can be generated by a single camera and 3D images should be generated by at least two cameras. Although the video-based activity monitoring technology is perceived as very intuitive, it still has some shortcomings. First, the video quality is susceptible to environmental light intensity and the range of viewing angle of a camera, so it is difficult to maintain satisfactory video quality at different times of the day. Besides, without concerning the privacy issue, users’ sensitive information can easily be leaked to attackers during video transmission in networks. Moreover, video storage and transmission require a lot of physical resources, for example, memory and bandwidth, thereby further limiting the wider application of this technology. Fortunately, the sensor-based activity monitoring technology can alleviate these problems to
some extent, so it is favored by most researchers and is more widely used in the field of activity recognition.

**Sensor-based activity monitoring.** With the emergence of a variety of low-cost sensors, sensor-based activity monitoring becomes more and more prevalent. According to different sensors used, this technology also falls into two categories: portable sensor-based and non-intrusive sensor-based. The former monitors user actions mainly based on RFID technology and acceleration sensors. RFID tags attached to objects can provide information of different objects. Therefore, it is easy to monitor user actions related to objects in the surrounding environment. Fox et al.\(^{11}\) design an RFID-based smart kitchen, in which RFID tags are attached to about 60 objects, for example, tablewares, coffee machines, cabinet doors, ovens, dishwashers, and refrigerators. In this application, user activities at different times of the day are collected and analyzed. In addition to RFID technology, acceleration sensors are also commonly used for activity monitoring. This kind of sensors is sensitive to activities that involve repetitive movements, for example, walking, running, standing, and climbing stairs. Zhang et al.\(^{12}\) acquire user activity data by attaching acceleration sensors to the hands of users, and then use a back-propagation (BP) neural network to analyze the sensor data for activity recognition.

However, the portable sensor-based technology still has some disadvantages. First, users are required to wear a series of sensors for most of the day, which may cause inconvenience to most users, especially the elderly. Second, some technical problems of portable sensors, for example, sensor size, wearing comfort, water resistance, and battery life, also limit the application of this technology to some extent. To alleviate these problems, smart phones are used for daily activity monitoring.\(^{13}\)

Compared with portable sensors, non-intrusive sensors do not impose any burden on users. They are usually low-cost and can be deployed at different positions in a smart home to record locations of users at any time. Then, the time and location data can be further used for activity recognition. Van Kasteren et al.\(^{14}\) establish a smart home environment with a variety of non-intrusive sensors, without interfering with users’ daily life. In this application, reed switch sensors are used to detect the door states of rooms, wardrobes, refrigerators, and ovens. Mercury contact sensors installed on objects such as medicine boxes, tablewares, and books are used to detect object movements. Floating sensors installed in the toilet can detect whether the user is using the toilet or the bathtub.

The Center for Advanced Studies in Adaptive Systems (CASAS) at Washington State University also build a smart home with non-intrusive sensors\(^{15,16}\) to enhance home-based medical technologies. In this project, passive infrared motion sensors are used to detect whether the target user appears in a certain area. Temperature sensors record the environmental temperature in real time. Object sensors monitor whether the objects are being used by the user. Water flow meters calculate the amount of water used by the user. The opening and closing status of doors are tracked by door sensors. Mobile phone use sensors detect whether the user is using a mobile phone.

As a result, the activity monitoring technology provides data support for the subsequent activity recognition, so it is an essential prerequisite for high-quality activity recognition. In the following subsection, we present some existing technologies of activity recognition.

**Activity recognition technology**

Sensor data collected by activity monitoring devices are then processed by specific algorithms to realize activity recognition. In this section, we introduce some typical activity recognition technology.

Generally, the collected sensor data can be seen as a time series. By dividing the time series, a series of fixed-length windows are obtained. Then, some statistical techniques are used to extract a feature vector from each time window. The most commonly used features for activity recognition include time and locations of sensor events, and the order of appearance of sensors in a window.\(^{17}\) Wu\(^{18}\) proposed a mixed feature extraction method based on time segment coding. Time segment is Gray-encoded and combined with other features to enrich the feature set and improve the recognition accuracy. In addition, some researchers combine environmental information (e.g. time, locations and traffic routes) of users with acceleration sensor data collected by smartphones,\(^{19}\) in order to enrich identifiable activity categories.

After feature extraction, a group of feature vectors should be manually labeled to build a training set for model training. Typical supervised model training techniques include template matching, discriminant, and generative methods. The template matching techniques calculate the distance between each pair of feature vectors and determine the activity category of a new feature vector according to the labels of its nearest neighbors.\(^{20,21}\) The discriminant techniques, which mainly depend on machine learning algorithms (e.g. artificial neural networks (ANNs) and decision trees), identify different activity categories by searching for boundaries between different categories of feature vectors. ANN\(^{22-24}\) mainly trains a complex network to model the non-linear relationships between feature vectors and activity categories. However, the high complexity of the network (objective function) usually makes the parameter tuning process time-consuming.
Moreover, the objective function is easy to fall into a local minimum, resulting in a poor ability of activity recognition. Therefore, it is important to design a reasonable network topology before model training. The decision tree algorithm continuously selects features that can best differentiate different activities according to information gain. The generative methods, for example, naive Bayes (NB) classifier and hidden Markov model, generally construct a joint probability distribution of feature vectors and labels, and then calculate the association probability of a new feature vector and different labels. Finally, the label with the highest probability is selected as the activity recognition result.

Based on the above analysis, we can see that the current activity recognition technologies in smart environments still have some shortcomings. First, activity recognition is mostly performed offline in existing works. However, in real-world applications, it remains a challenge to realize real-time activity recognition based on the online streaming sensor data. Second, sensor data cannot be divided into different segments according to manually assigned class labels, since manual labeling in real-time environments is impossible. In addition, when processing the latest sensor data in real-time applications, only historical data are available, while future data are not available, which is very different from offline data processing techniques.

In this work, we propose a novel approach for online real-time activity recognition in smart homes. This approach only depends on historical data when analyzing the latest sensor data, so it is suitable for online real-time applications. Moreover, this approach adopts an advanced feature vector extraction technique enhanced by mutual information between sensors, which can effectively reduce the impacts of irrelevant information contained in a window of sensor events and further improve the accuracy of activity recognition. Finally, a genetic algorithm–optimized multi-class SVM classifier is used to realize activity recognition. Here, the genetic algorithm can automatically select optimal hyperparameters for the SVM classifier, thereby reducing the inaccuracy and inefficiency caused by manual tuning of hyperparameters.

Preliminaries

This section first introduces the SVM theory for classification, and then presents the problem statement and an overview of the proposed approach for activity recognition.

Introduction to SVMs

SVM is a popular machine learning algorithm that provides solutions for classification and regression problems. Here, we mainly focus on the classification problem. Given a group of training samples, SVM aims to find the training cases that lie on the class boundaries, that is, the support vectors. These support vectors can determine an optimal separating hyperplane (OSH) between different classes. In other words, only the training cases that lie on the class boundaries are necessary for discrimination and other training samples can be discarded.

Suppose that a training set of \( N \) cases is denoted by \( \{x_i, y_i\} (i = 1, \ldots, N) \), where \( x_i \) is a \( q \)-dimensional vector and \( y_i \in \{-1, +1\} \) is the corresponding class label. An OSH aims to maximize the margin between different classes and is represented by \( w \cdot x + b = 0 \), where \( w \) is a vector perpendicular to the hyperplane, \( x \) is a point lying on the hyperplane, and \( b \) is the bias. If the training cases are linearly separable, the support vectors that determine the class boundary can be denoted by

\[
y_i(w \cdot x_i + b) = 1
\]

SVM aims to maximize the discrimination margin \( 2/\|w\| \), which is determined by the distance between the hyperplane and the training samples nearest to it. Maximizing the margin is equivalent to solving the following optimization problem

\[
\min_{w, b} \left\{ \frac{1}{2} \|w\|^2 \right\}
\]

under the constraint denoted by \( y_i(w \cdot x_i + b) \geq 1 \). Solving this quadratic problem yields the hyperplane parameter as follows

\[
w^* = \sum_{\forall x_i \in S} \alpha_i y_i x_i
\]

and

\[
b^* = y_j - \sum_{\forall x_j \in S} \alpha_j y_j (x_i \cdot x_j)
\]

where \( S \) denotes the set of support vectors for both classes, and \( \alpha_i \) is a trained weight on the corresponding support vector \( x_i \), the subscript \( j \) satisfies \( \alpha_j > 0 \). Based on the solution, an arbitrary new sample \( x \) can be classified by the following function

\[
f(x) = \text{sign}(w^* \cdot x + b^*) = \text{sign} \left( \sum_{\forall x_i \in S} \alpha_i y_i x_i \cdot x + b^* \right)
\]
high-dimensional feature space $H$, that is, $\phi: \mathbb{R}^q \rightarrow H$. Therefore, $\phi(x)$ in the high-dimensional space $H$ can represent the original training sample $x$. In addition, a positive definite kernel $k(x,x_i)$ can be used to reduce the costly computation of $\phi(x) \cdot \phi(x_i)$ in the high-dimensional space $H$

$$\phi(x) \cdot \phi(x_i) = k(x,x_i)$$

thereby leading to the following decision function

$$f(x) = \text{sign} \left( \sum_{x_i \in S} \alpha_i y_i k(x,x_i) + b \right)$$

The commonly used kernel functions\textsuperscript{36} include linear kernel, radial basis function (RBF) kernel, polynomial kernel, and sigmoid kernel.

SVMs were originally designed for binary classification, but can be extended for multi-class classification. A multi-class classification problem should be first reduced to a group of binary classification problems, then the basic SVM approach can be applied. There are generally two main approaches for multi-class classification: “one-against-all” and “one-against-one.” The former trains a group of binary SVM classifiers, each separating one class from the rest. In this approach, $n$ decision functions should be built for each sample $x$ if there are $n$ classes in total. Each decision function indicates whether the case $x$ belongs to the class under consideration or not. Finally, the case is allocated to the class for which the function $\sum_{x_i \in S} \alpha_i y_i k(x,x_i) + b$ has the largest value. The second approach trains a binary classifier for each pair of classes, thereby yielding a total of $n(n-1)/2$ binary classifiers. The class that receives the maximum number of votes is selected as the final category of a sample $x$.

Problem statement and approach overview

**Problem statement.** We aim to realize online real-time activity recognition in health smart homes using an optimized SVM classifier. Suppose that a total of $M$ sensors are deployed in a smart home to monitor human activities and a series of sensor events $\{e_i\} (i = 1, 2, \ldots)$ is collected over time. Each sensor event is formalized as a quintuple $e_i = (d_i, t_i, s_i, l_i, a_i)$, where $d_i$, $t_i$, $s_i$, $l_i$, and $a_i$ denote the date, time, sensor, location, and activity category corresponding to the event $e_i$. Please note that the activity categories are not monitored information, but need to be manually annotated or predicted. Manually labeled sensor events are used as training data to train an SVM-based activity recognition model, which can be used to classify newly generated sensor events in real time.

**Approach overview.** In order to address the problem stated above, we propose a novel activity recognition approach based on genetic algorithm–optimized SVM. Procedures involved in this approach are presented in Figure 1. First, the collected sensor events are divided into a sequence of overlapping sliding windows. Then, a feature vector amended by mutual information between sensors is extracted from each sliding window. Afterward, a group of feature vectors with manually labeled categories are used to train a multi-class SVM classifier, and the classification accuracy of the trained model is tested on a test set. In addition, the genetic algorithm is employed to automatically select optimal hyperparameters for the SVM model. Finally, the trained model is used to classify newly generated sensor events to realize real-time activity recognition.

![Figure 1. Approach overview.](image-url)
Activity recognition based on genetic algorithm–optimized SVM classifier

In this section, we elaborate on the proposed activity recognition approach, mainly including sensor event sequence segmentation, feature extraction, and genetic algorithm–optimized SVM for activity recognition.

Sensor event sequence segmentation

In order to build a training set, the collected sensor event sequence should be first segmented into a series of fixed-length overlapping sliding windows from which feature vectors are extracted. Formally, a sequence of \( N \) sensor events is denoted by \( \{ e_1, e_2, \ldots, e_N \} \). Then, it is segmented into a series of sliding windows \( \{ W_1, W_2, \ldots, W_K \} \), where \( W_i (i \in \{ 1, \ldots, K \}) \) is the window consisting of \( L \) consecutive sensor events from \( e_{i-L+1} \) to \( e_i \). In the window \( W_i \), the class label of the last event \( e_i \) is used as the label of \( W_i \), and the event subsequence \( \{ e_{i-L+1}, \ldots, e_{i-1} \} \) is regarded as the preceding context of the last event \( e_i \). In addition, the window length \( L \) is mainly determined by the average number of sensor events contained in different activities. Choosing an appropriate value of \( L \) can effectively reduce the adverse impact of irrelevant information contained in the current window. The window lengths of different activities are investigated and shown in Table 1, and the average window length is 32.89. For simplicity, \( L \) is set to 30.

Feature extraction

After obtaining a set of sliding windows, we can extract a feature vector from each window. Activity recognition aims to build a mapping relationship between feature vectors and activity categories, so extracting appropriate feature vectors is essential for high-quality activity recognition. Traditionally, the number of occurrences of each sensor in a window is regarded as an important feature. However, this feature extraction technique has an obvious disadvantage. When a window contains one or more activity transitions, most sensor events in the preceding context are irrelevant to the last sensor event, so a lot of irrelevant sensor information will be incorporated into this feature. Figure 2 illustrates a sample window containing two activity transitions from “Leaving” to “Other Activity” and from “Other Activity” to “Toilet.” Please note that the class label of the last sensor event in the window is “Toilet,” so all the other sensor events in the window are considered as the preceding context of the “Toilet” activity. However, most sensor events in the window belong to the “Leaving” or the “Other Activity” categories, both of which are irrelevant to the “Toilet” activity. Therefore, using the number of occurrences of each sensor as a feature is likely to result in misclassification of the last sensor event.

In order to alleviate this problem, we use mutual information between sensors to amend the number of occurrences of each sensor.\(^{37,38}\) Mutual information measures the interdependence between two random variables. Here, the mutual information between two sensors is defined as the probability that the two sensors appear next to each other in the entire sensor event sequence. Supposing that there are \( N \) sensor events in the entire sequence, mutual information \( MI(i,j) \) between the two sensors \( s_i \) and \( s_j \) is defined by

\[
MI(i,j) = \frac{1}{N} \sum_{k=1}^{N-1} \delta(e_k, s_i) \delta(e_{k+1}, s_j)
\]

(8)

where \( \delta(e_k, s_i) \) is a binary value indicating whether the sensor \( s_i \) appears in the sensor event \( e_k \), and is formally denoted by

\[
\delta(e_k, s_i) = \begin{cases} 
1, & \text{if } s_i \text{ appears in } e_k \\
0, & \text{otherwise}
\end{cases}
\]

(9)

If two sensors frequently appear next to each other, their mutual information is relatively high. Therefore, the mutual information between sensors can be used to measure the correlation between a preceding sensor event and the last sensor event in a sliding window. Accordingly, the amended feature vector \( x_i \) of the window \( W_i \) can be denoted by

\[
x_i = (w_1, w_2, \ldots, w_M, t_s, t_e, id, w_b)
\]

(10)

where \( M \) is the total number of sensors deployed in a smart environment; \( id \) is the index of the sensor that appears in the last event of the window; \( t_s \) and \( t_e \) (\( 0 \leq t_s, t_e \leq 24 \)) are, respectively, the start and end time

| Behavior          | The average window length | The minimum window length | The maximum window length |
|-------------------|---------------------------|---------------------------|---------------------------|
| Bathe             | 16.67                     | 5                         | 36                        |
| Bed_Toilet_       | 15.35                     | 2                         | 113                       |
| Transition        | 62.75                     | 7                         | 195                       |
| Cook              | 19.41                     | 4                         | 80                        |
| Leaving           | 43.15                     | 9                         | 218                       |
| Personal_Hygiene  | 31.53                     | 5                         | 98                        |
| Wash_Dishes       | 39.89                     | 22                        | 130                       |
| Work_Ac_Table     | 35.89                     | 3                         | 526                       |
| Average           | 32.89                     | 7.5                       | 179.2                     |

Table 1. Window lengths of different activities.
of $W_i$; and $w_m (m \in \{1, 2, \ldots, M\})$ denotes the amended number of occurrences of sensor $s_m$ in $W_i$, and is formally defined by

$$w_m = o_m \times MI(m, id)$$

(11)

where $o_m$ is the actual number of occurrences of sensor $s_m$ in $W_i$, and $MI(m, id)$ is the mutual information between $s_m$ and $s_{id}$. Generally, there exists a significant difference between activities performed on weekends and weekdays, so a binary feature $w$ is used to indicate whether the current window occurs on a weekend. In addition, there may exist some causal relationships between two adjacent activities, so the feature $b_p$ denotes the activity category of the preceding window of $W_i$, which has no overlap with $W_i$. The feature vectors extracted from the training set, along with their manually labeled categories, are used to train a multi-class SVM classifier.

### Genetic algorithm–optimized SVM for activity recognition

The SVM model has four hyperparameters: kernel function, penalty coefficient, kernel function coefficient, and polynomial degree. The last parameter is only useful when a polynomial kernel function is used. Different parameter configurations will undoubtedly affect the accuracy of the activity recognition model. Therefore, the genetic algorithm is used to automatically select the optimal hyperparameters for the SVM classifier.

The genetic algorithm is a well-known heuristic search algorithm, which is good at handling large-scale and high-dimensional search problems. It starts with selecting the best or fittest individuals from an initial population, where each individual represents a candidate solution to the selection problem. Then, the reproduction, crossover, and mutation operators are performed on the selected individuals to produce offsprings with higher fitness values. This process iterates over the search space until a termination condition is met. In this work, the genetic algorithm is used to select optimal hyperparameters for the SVM model.

Before performing the genetic algorithm, each hyperparameter should be encoded by a consecutive series of binary genes on a chromosome. For example, the hyperparameter “kernel function” occupies two consecutive genes, which can encode four different values corresponding to the four candidate kernel functions. Then, the initial population is randomly generated and the fitness value of each chromosome is calculated by

$$fitness = \frac{macroF_1 - \min}{\max - \min + c}$$

(12)

where $macroF_1$ is the macro-$F_1$-score obtained when performing the SVM classifier with the current hyperparameters on the test set (the definition of macro-$F_1$-score is presented in section); $\max$ and $\min$ are, respectively, the maximum and minimum macro-$F_1$-scores of solutions in the current generation; and $c$ is a small positive value (e.g. $c = 0.001$) used to prevent the denominator from being 0 when a generation converges, that is, $\max$ is equal to $\min$.

As mentioned above, the genetic algorithm includes three fundamental operators: reproduction, crossover, and mutation. The reproduction operator copies chromosomes with highest fitness values from the current generation to the next one. It is an elite strategy aimed at producing better solutions for the next generation based on the high-quality chromosomes of the current generation. Crossover is the most important operator. It first selects chromosomes with the highest fitness values as parents and then swap the parents’ sections after a selected crossover point to produce two offsprings. The mutation operator can ensure diversification of solutions and avoid a local optimum by randomly

Figure 2. A sample window containing activity transitions.

| Date       | Time          | Sensor | Location    | Behavior     |
|------------|---------------|--------|-------------|--------------|
| 2012-7-20  | 13:04-54.128796 | D002   | OutsideDoor | Leaving      |
| 2012-7-20  | 13:04-54.281891 | M001   | OutsideDoor | Leaving      |
| ...        | ...           | ...    | ...         | ...          |
| 2012-7-20  | 14:40-50.094331 | M001   | OutsideDoor | Leaving      |
| 2012-7-20  | 14:40-50.622821 | M010   | LivingRoom  | OtherActivity|
| 2012-7-20  | 14:40-51.758644 | M001   | OutsideDoor | OtherActivity|
| ...        | ...           | ...    | ...         | ...          |
| 2012-7-20  | 14:41-00.008181 | M011   | Bedroom     | OtherActivity|
| 2012-7-20  | 14:41-02.782050 | MA015  | Bathroom    | Toilet       |
| 2012-7-20  | 14:41-12.732782 | MA015  | Bathroom    | Toilet       |
changing some genes on a chromosome. The three operators are repeated for many generations until a termination condition is satisfied. Once the optimal hyper-parameters are determined, the corresponding SVM classifier can be used for activity recognition.

Experiments and discussion

In this section, we conduct a series of experiments on a real-world data set to validate the effectiveness of the proposed activity recognition approach.

Data set

We use a freely available data set “Human Activity Recognition from Continuous Ambient Sensor Data” (https://archive.ics.uci.edu/ml/datasets.php) provided by the Center of Advanced Studies in Adaptive Systems (CASAS) at Washington State University for experiments. The entire data set records the daily activities of 15 volunteers in 15 smart homes within a month, resulting in 15 sub-data sets: CSH101–CSH115. In the CASAS project, motion sensors, door sensors, light sensors, temperature sensors, and other kinds of sensors are deployed in locations throughout a smart home for activity and environment monitoring. In our experiments, for simplicity, only the data generated by motion sensors and door sensors are used.

Data preprocessing

We analyzed the common activities in the 15 data sets, then merged some similar activities and finally got 11 activity categories. For example, the three activities “Eating Breakfast,” “Eating lunch,” and “Eating dinner” can be merged into the “Eat” activity. The final 11 activity categories are “Bathe,” “Bed_Toilet_Transition,” “Cook,” “Eat,” “Leaving,” “Personal_Hygiene,” “Sleep,” “Toilet,” “Wash_Dishes,” “Work_At_Table,” and “Other_Activity.” The distribution of different activities in the first data set CSH101 is shown in Figure 3. Obviously, the distribution of different categories is not uniform, so a robust SVM method which treats unbalanced cases based on the weights of classes is employed.

Evaluation metric

In our experiments, we use three metrics: macro-precision $\text{macroP}$, macro-recall $\text{macroR}$, and macro-F1-score $\text{macroF}$, to evaluate the performance of the proposed approach. $\text{macroP}$ denotes an average per-class agreement of the actual class labels with those predicted by a classifier. $\text{macroR}$ represents an average per-class effectiveness of a classifier to identify class labels. $\text{macroF}_1$ is a harmonic mean of $\text{macroP}$ and $\text{macroR}$. The three metrics are formally denoted by

\[
\text{macroP} = \frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fp_i}}{n}
\]

\[
\text{macroR} = \frac{\sum_{i=1}^{n} \frac{tn_i}{tn_i + fn_i}}{n}
\]

\[
\text{macroF}_1 = \frac{2 \times \text{macroP} \times \text{macroR}}{\text{macroP} + \text{macroR}}
\]

where $n$ is the total number of activity categories, $tp_i (i = 1, 2, \ldots, n)$ is true positive for category $c_i$, $tn_i$ is true negative, $fp_i$ is false positive, and $fn_i$ is false negative, respectively.

Performance comparison

In order to evaluate the performance of the proposed approach, the cross-validation technique is used. Specifically, 70% of the data set is randomly selected for model training and the rest for testing. This procedure repeats 10 times and the average results are reported.

First, in order to verify the effectiveness of the sensor mutual information-amended feature extraction technique, the performances ($\text{macroP}$, $\text{macroR}$, and $\text{macroF}_1$) of different feature extraction methods are compared and the results are illustrated in Figure 4. The $f_1$ method refers to a method that only uses binary variables as features to indicate whether the sensors
appear in a window. The \( f_2 \) method uses the numbers of occurrences of sensors in a window as features. The \( f_3 \) (or \( f_4 \)) method combines the features used by \( f_1 \) (or \( f_2 \)) and the \( t_c, t_e, id, w, \) and \( b_p \) features mentioned in section. The experimental results show that \( f_1 \) yields the worst performance. Compared with \( f_1 \), the performance of \( f_2 \) is improved, which indicates that the number of occurrences of a sensor is a better feature than the binary value indicating whether a sensor appears. The \( f_3 \) and \( f_4 \) methods integrate more useful features based on the first two methods, so the performances of the two methods are further improved. The \( f_2 \) method uses feature vectors amended by mutual information between sensors, and it achieves the best performance. The experimental results prove that the feature extraction method proposed in this article is superior to traditional feature extraction techniques in terms of activity recognition accuracy.

When performing the genetic algorithm for automatic hyperparameter selection, 60\% of individuals in each generation are selected as parents, and the mutation probability is set to 1\%. The population size of each generation is set to 20 and the maximum number of iterations is set to 20. We performed the genetic algorithm on the CSH101 data set as an example. The average fitness value of each generation is depicted in Figure 5. The fitness curve tends to be stable after 12 iterations. We consider that the genetic algorithm has found the optimal hyperparameters of the SVM model. The corresponding chromosome with the highest fitness value in the final generation represents the optimal hyperparameters of the SVM classifier.

Figure 6 illustrates the confusion matrix of the SVM model for activity recognition on the CSH101 data set. The \( x \)-axis and the \( y \)-axis denote the predicted and the actual activity class labels, respectively. The saturation of the \((i,j)\)th element of the matrix represents the proportion of samples that actually belong to class \( y_j \) but are classified into class \( y_i \). The diagonal elements represent all correctly classified cases. The high saturation of the diagonal elements indicates that the SVM model can correctly classify most daily activities.

In addition, the precisions, recalls, and \( F_1 \)-scores obtained on different activity classes are shown in Table 2. The last two rows, respectively, report the arithmetic average and the weighted average of the activity recognition performance on different classes. The weighted average is calculated based on the weight of each class, which is inversely proportional to the number of samples in this class. We can see that the
classifier yields a relatively high recall but a low precision on the “Sleep” class, which means that most “Sleep”-labeled samples are correctly classified, but some samples that actually belong to other classes are also classified into the “Sleep” class. Moreover, the classifier yields the minimum precision, recall and $F_1$-score on the “Leaving” category. In other words, a relatively large proportion of samples with the “Leaving” label are incorrectly classified into other classes and many samples from other classes are also incorrectly classified into the “Leaving” class. The recall values on the “Bed_Toilet_Transition,” “Eat,” and “Work_At_Table” classes are all 100%, which means that all samples from the three categories are correctly classified. The $F_1$-scores on most classes (except the “Leaving” class) are above 0.9, which means that the SVM classifier has an excellent ability of activity recognition.

Next, in order to investigate the impact of the training set size on the generalization ability of the SVM classifier, we conducted experiments on a series of training sets of different sizes. The corresponding experimental results are reported in Figure 7. The $x$-axis denotes the ratio of training samples randomly selected from the CSH101 data set, and the $y$-axis represents the macro-$F_1$-scores of the trained classifier. When the training set is very small compared to the test set, the trained model performs well on the training set but performs poorly on the test set. In this case, the SVM model is overfitted and has a poor generalization ability. As the training set size increases, the macro-$F_1$-score on the training set remains high, and the macro-$F_1$-score on the test set gradually increases and tends to be stable. The results show that appropriately increasing the size of the training set can effectively alleviate the overfitting problem and improve the generalization ability of the SVM classifier.
In addition, the impacts of the training set size on the model training time and the recognition accuracy on the test set are depicted in Figure 8. The training time increases approximately linearly with the size of the training set. The macro-$F_1$-score on the test set tends to be stable when the ratio of training samples to the entire data set exceeds 70%. Therefore, choosing an appropriate size of the training set can help the SVM classifier strike a balance between training time and classification accuracy.

Figure 9 shows the performance comparison of five different activity recognition models: the genetic algorithm–optimized SVM proposed in this work, back-propagation–artificial neural network (BP-ANN), logistic regression (LR), NB, and C4.5 decision tree. BP-ANN employs a classic three-layer BP neural network. LR uses a multi-class logistic regression model based on the “one vs one” strategy. NB is based on a Gaussian model. C4.5 is a decision tree–based classification algorithm. The experimental results show that the classification accuracy of SVM is significantly better than that of LR, NB, and C4.5. BP-ANN and SVM yield comparable classification accuracy, but the training time of BP-ANN is usually longer than that of SVM.

Figure 10 reports the macro-$F_1$-scores of the proposed approach on the 10 data sets: CSH101–CSH110. We can see that all macro-$F_1$-scores on the 10 data sets are above 0.9, which means that the proposed approach has an excellent ability of activity recognition on different data sets.

**Conclusion**

In this work, we proposed a real-time activity recognition approach based on the genetic algorithm–optimized SVM classifier. Mutual information between sensors is utilized to amend feature vectors, thereby reducing the impact of irrelevant information contained in a sliding window of sensor events and further improving the accuracy of activity recognition. In addition, the SVM classifier is enhanced by the genetic algorithm for automatic hyperparameter selection, thereby avoiding the costly manual selection of hyperparameters. This approach can realize high-quality real-time activity recognition for elderly people in smart home environments and allow them to live more safely and independently at home.
However, the work presented in this article still has some limitations. First, training an SVM classifier requires a large number of labeled data samples, which makes manual labeling a costly process. In addition, each smart home has to train a specific SVM classifier for activity recognition, since people in different environments have different activity patterns. In other words, it is difficult to share a common activity recognition model between different smart environments. In the future work, we will try to use transfer learning technology to solve these problems, so as to realize knowledge sharing between different environments, reduce the burden of manual labeling and further improve the efficiency of activity recognition.

Acknowledgements
The authors appreciate the reviewers for their helpful comments and suggestions for the improvement of this paper.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported, in part, by the National Natural Science Foundation of China under grant 61802016 and 61702506; in part, by the Interdisciplinary Research Project for Young Teachers of USTB (Fundamental Research Funds for the Central Universities) under grant FRF-IDR-Y-19-016; in part, by National Key Research and Development Project under grant 2017YFB0802805; and in part, by the National Social Science Foundation of China under grant 17ZDA331.

ORCID iD
Yan Hu https://orcid.org/0000-0002-1712-2771

References
1. Fleury A, Vacher M and Noury N. SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. *IEEE T Inform Tech Biomed* 2010; 14(2): 274–283.
2. Chan M. A review of smart homes-present state and future challenges. *Comput Method Prog Biomed* 2008; 91(1): 55–81.
3. Intille SS. A new research challenge: persuasive technology to motivate healthy aging. *IEEE T Inform Tech Biomed* 2004; 8(3): 235–237.
4. Abowd GD, Mynatt ED and Rodden T. The human experience [of ubiquitous computing]. *IEEE Pervas Comput* 2002; 1(1): 48–57.
5. Lebellego G, Noury N, Virone G, et al. A model for the measurement of patient activity in a hospital suite. *IEEE T Inform Tech Biomed* 2006; 10(1): 92–99.
6. Bonhomme S, Campo E, Estède D, et al. Prosafe-extended, a telemedicine platform to contribute to medical diagnosis. *J Telemed Telecare* 2008; 14(3): p116–119.
7. Poppe R. A survey on vision-based human action recognition. *Image Vision Comput* 2010; 28(6): 976–990.
8. Wu C, Khalili AH and Aghajan HK. Multiview activity recognition in smart homes with spatio-temporal features. In: *Proceedings of the 2010 4th ACM/IEEE international conference on distributed smart cameras (ICDSC)*, Atlanta, GA, 31 August–4 September 2010, pp.142–149. New York: ACM.
9. Chang MC, Krahnstoever N, Lim S, et al. Group level activity recognition in crowded environments across multiple cameras. In: *Proceedings of 2010 7th IEEE international conference on advanced video and signal based surveillance (AVSS)*, Boston, MA, 29 August–1 September 2010, pp.56–63. New York: IEEE.
10. Klasnja PV, Consolvo S, Choudhury T, et al. Exploring privacy concerns about personal sensing. In: *Proceedings of the 2009 7th international conference on pervasive computing (Pervasive)*, Nara, Japan, 11–14 May 2009, pp.176–183. New York: IEEE.
11. Patterson D, Fox D, Kautz H, et al. Fine-grained activity recognition by aggregating abstract object usage. In: *Proceedings of the 2005 9th IEEE international symposium on wearable computers (ISWC)*, Osaka, Japan, 18–21 October 2005, pp.44–51. New York: IEEE.
12. Zhang L, Kuang Z, Li K, et al. Human activity behavior recognition based on acceleration sensor and neural network. *Modern Electr Tech* 2019; 12(7): 71–7478.
13. Zhang J, Wu Z, Li F, et al. Attention-based convolutional and recurrent neural networks for driving behavior recognition using smartphone sensor data. *IEEE Access* 2019; 7: 148031–148046.
14. Van Kasteren T, Noulas AK, Englebienne G, et al. Accurate activity recognition in a home setting. In: *Proceedings of the 2008 10th international conference on ubiquitous computing (Ubicomp)*, Seoul, 21–24 September 2008, pp.1–9. New York: IEEE.
15. Cook D. Learning setting-generalized activity models for smart spaces. *IEEE Intel Syst* 2012; 27(1): 32–38.
16. Cook D, Krishnan N and Rashidi P. Activity discovery and activity recognition: a new partnership. *IEEE T Syst Man Cyb Part B* 2013; 43(3): 820–828.
17. Tong Y. Research on activity recognition in smart home based on conditional random fields. PhD thesis, Dalian Maritime University, Dalian, China, 2015.
18. Wu J. Smart home activity recognition and behavior analysis based on environmental sensors. PhD thesis, Xiamen University, Xiamen, China, 2016.
19. Bettini C, Civitarese G and Presotto R. Caviar: context-driven active and incremental activity recognition. *Knowl Based Syst* 2020; 196: 105816.
20. Yan X. Weighted KNN classification algorithm based on mean distance of category. *Comput Syst Appl* 2014; 23(2): 128–132.
21. Du C. Railway engineering staff behavior recognition method based on machine learning. *Comput Syst Appl* 2019; 28(7): 199–205.

22. Mehr HD, Polat H and Cetin A. Resident activity recognition in smart homes by using artificial neural networks. In: 4th international Istanbul smart grid congress and fair (ICSG), Istanbul, 20–21 April 2016. New York: IEEE.

23. Suto J and Oniga S. Efficiency investigation of artificial neural networks in human activity recognition. *J Ambient Intel Human Comput* 2018; 9(4): 1049–1060.

24. Oniga S and Suto J. Activity recognition in adaptive assistive systems using artificial neural networks. *Elektrotechnika Ir Elektrotechnika* 2016; 22(1): 68–72.

25. Lee YS and Cho SB. Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data. *Neurocomputing* 2014; 126: 106–115.

26. Fan L, Wang Z and Wang H. Human activity recognition model based on decision tree. In: *Proceeding of 1st international conference on advanced cloud and big data (CBD)*, Nanjing, China, 13–15 December 2013, pp.64–68. New York: IEEE.

27. Prosegger M and Bouchachia A. Multi-resident activity recognition using incremental decision trees. New York: Springer, 2014.

28. Yan He ZW and Wang K. A user-independent behavior recognition model based on multi-classifier fusion. *J Xi’an Univ Post Telecommun* 2016; 21(5): 50–54.

29. Yang X, Dinh A and Chen L. Implementation of a wearable real-time system for physical activity recognition based on Naive Bayes classifier. In: *Proceedings of the 2010 2nd international conference on bioinformatics and biomedical technology (ICBBT)*, Chengdu, China, 16–18 April 2010, pp.101–105. New York: IEEE.

30. Trabelsi D, Mohammed S, Chamrouki F, et al. An unsupervised approach for automatic activity recognition based on hidden Markov model regression. *IEEE T Automat Sci Eng* 2013; 10(3): 829–835.

31. Huang C, Davis LS and Townshend JRG. An assessment of support vector machines for land cover classification. *Int J Remote Sens* 2002; 23(4): 725–749.

32. Vapnik VN. *The nature of statistical learning theory*. New York: Springer, 2000.

33. Muller K, Mika S, Ratsch G, et al. An introduction to kernel-based learning algorithms. *IEEE T Neural Netw* 2001; 12(2): 181.

34. Melgani F and Bruzzone L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE T Geosci Remote Sens* 2004; 42(8): 1778–1790.

35. Mathur A and Foody GM. A relative evaluation of multiclass image classification by support vector machine. *IEEE T Geosci Remote Sens* 2004; 42(6): 1335–1343.

36. Trivedi SK, Dey S, Trivedi SK, et al. Effect of various kernels and feature selection methods on SVM performance for detecting email spams. *Int J Comput Appl* 2013; 66(21): 18–23.

37. Gong L. *Research on user daily behavior recognition algorithms for smart home*. PhD thesis, Chongqing University of Posts and Telecommunications, Chongqing, China, 2018.

38. Guyon I and Elisseeff A. An introduction to variable and feature selection. *J Mach Learn Res* 2003; 3(6): 1157–1182.

39. Goldberg DE. *Genetic algorithm in search, optimization and machine learning*. Boston, MA: Addison Wesley, 1989, p.372.

40. Jung KM. Support vector machines for unbalanced multiclassification. *Math Probl Eng* 2015; 2015(pt2): 2949851–2949857.

41. Sokolova M and Lapalme G. A systematic analysis of performance measures for classification tasks. *Inform Proces Manag* 2009; 45(4): 427–437.