IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 19, 2018, accepted August 26, 2018, date of publication September 10, 2018, date of current version September 28, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2868984*

# An Unsupervised User Behavior Prediction Algorithm Based on Machine Learning and Neural Network For Smart Home

**TIANKAI LIANG[1], BI ZENG[1], JIANQI LIU[2], (Member, IEEE), LINFENG YE[1], AND CAIFENG ZOU[3]**

[1]School of Computer, Guangdong University of Technology, Guangzhou 510006, China
[2]School of Automation, Guangdong University of Technology, Guangzhou 510006, China
[3]Guangdong Mechanical and Electrical College, Guangzhou 510515, China

Corresponding author: Jianqi Liu (liujianqi@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701122, in part by the Science and Technology Plan Project of Guangdong Province, China, under Grant 2016B010108004, in part by the Natural Science Foundations of Guangdong Province, China, under Grant 2016A030313734 and Grant 2017A030313431, in part by the Science and Technology Program of Guangzhou, China, under Grant 201804010238, in part by the Application Major Project of Guangdong Province under Grant 201604020016, and in part by the Special Project of Industry-University-Institute Cooperation of Guangdong Province under Grant 2014B090904080.

**ABSTRACT** The user operates the smart home devices year in year out, have produced mass operation data, but these data do not be utilized well in past. Nowadays, these data can be used to predict user's behavior custom with the development of big data and machine learning technologies, and then the prediction results can be employed to enhance the intelligence of a smart home system. In view of this, this paper proposes a novel unsupervised user behavior prediction (UUBP) algorithm, which employs an artificial neural network and proposes a forgetting factor to overcome the shortcomings of the previous prediction algorithm. This algorithm has a high-level of autonomous and self-organizing learning ability while does not require too much human intervention. Furthermore, the algorithm can better avoid the influence of user's infrequent and out-of-date operation records, because of the forgetting factor. Finally, the use of real end user's operation records to demonstrate that UUBP algorithm has a better level of performance than other algorithms from effectiveness.

**INDEX TERMS** Smart home, behaviors prediction, data mining, machine learning, unsupervised learning, personalized recommendation.

## I. INTRODUCTION

With the development of new generation wireless communications and big data technologies, great changes have taken place in some industries [1], such as intelligent transportation, and Internet of things (IoT) [2], [3]. The Smart Home is a typical application in IoT and ubiquitous computing in which the house environment is monitored by ambient intelligence to provide context-aware services and facilitate remote home control [4]. The features of a typical smart home are that the devices are interconnected by the network including wired and wireless communication technology and controlled by a smart terminal such as a mobile phone or a personal computer through the Internet [5]. The framework of a smart home system is shown in Figure 1.

In a smart home, user can operate the home devices at one's own will, which provides the convenience by controlling their living environment. For example, people can start the air conditioner in advance by smart home system, before they commence their journey home. When they enter their home, the indoor temperature has become pleasant. The existing smart home has achieved remote or automatic control to some extent and is able to promote the safety, and comfort of a house [6], [7]. However, how to design a smart home system with the features of an intelligent decision and autonomous control still have a long way to go.

The intelligence level of an existing smart home system can be divided into three layers in summary [8]. Firstly, the remote operation in a smart home without intelligent decision layer is low-level intelligence that is far away from automation or intelligent control. The user transmits the control messages through the mobile application (APP) to operate a remote control. For example, when it rains, the

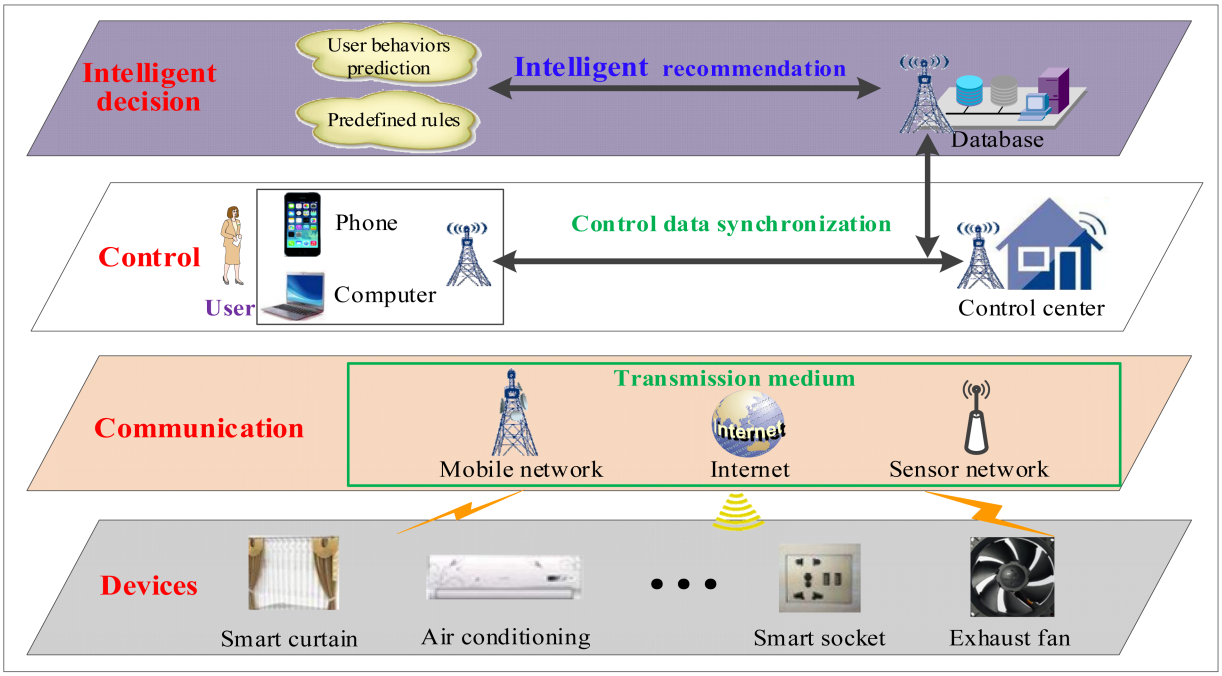


**FIGURE 1.** Framework of smart home system.

user presses the icon of the 'closing window' within the APP. Consequently, the ''close window'' command is sent to the smart home controller, and then the controller closes the window by executing this command. This is a typical remote operation method and cannot be completed without manual control by the user. Secondly, the mid-level intelligent smart home depends on intelligent environment-aware devices to collect the environmental information and subsequently make the proper response according to a predefined set of rules. For example, when it rains, the raindrop sensors perceive the change, make the decision by multi-sensors data fusion algorithm and execute the decision by the actuator. This intelligent control is able to realize self-adaptive control when conditions within the environment are changed. Thirdly, high-level intelligence of a smart home will offer the personalized service and friendly user experience. This solution not only adapts to the change of environment but also adapts user behaviors by itself. Therefore, it needs to possess the learning ability and can then predict user behaviors, and will over time evolve into seamless companions for the user [9]. Nowadays, the majority of existing smart home systems remain in the category of mid-level intelligence.

Recently, with the advance of the big data technology and artificial intelligence algorithms such as Neural Network (NN), K-means and Density-based Spatial Clustering of Applications with Noise (DBSCAN), the more personalized service which can be improved by the user experience. These improvements generate additional business value and can be offered by mining the mass historical data and discovering valuable rules. For example, Wei *et al.* [10] proposed a segmentation strategy of customer behaviors based on K-means to predict the consumer behaviors in the Chinese mobile communication market based on the consumer's consumption records. This strategy can effectively recommend the corresponding telecom packages according to the user's preference, increasing the income and competitiveness of the business. If these technologies can be applied to improve intelligence within smart homes, the market outlook of the smart home has the potential to achieve a successful breakthrough. Therefore, a number of researchers focus attention to predict user behaviors, derived from mass historical and real-time user operation records data, aiming to improve the intelligence of the smart home [11].

However, there are some problems which need to address in user behaviors prediction algorithms. Firstly, over how much manual intervention is required during the initialization of an algorithm. For example, some algorithms such as K-means algorithm or other K-means-based algorithms need upon clustering number when it tries to predict $k$ behaviors of a user by partitioning the user operation records into $k$ clusters [12]. Therefore, during the initialization of those algorithms, there is a need to give a clustering number $k$ according to the past experiences, but the subjective experiences may not express the practical partition fully, which ultimately influences the performance of the prediction algorithm [13]. Secondly, the low ability to distinguish the importance of user operation records is an issue which requires attention. User behaviors will change dynamically over periods of time. So, the prediction algorithm is needed to know that recent operation records are more important than the

out-of-date records. The regulation of importance and the ability to slowly forget the out-of-date records like a human being does is crucial [14]. Therefore, the proposition of a feasible and novel algorithm to solve those problems and at the same time, predict user behaviors more intelligently is still a major challenge in the domain of smart home.

In this paper, it is proposed that the UUBP algorithm based on machine learning and neural network to mine the massive user behaviors data is a potential solution. This aims at enhancing the intelligence level in existing smart home systems. The major contributions of this paper are as follows:

- In order to decrease the manual error in the initialization stage of the UUBP algorithm, an initializing learner, ANN, is proposed to get the user behavior number self-organized. The initialization approach is based on a neural network and is more impersonal than manual selection. Furthermore, compared with those similar algorithms which do not need upon clustering number, such as DBSCAN, it has a higher level of effectiveness.
- An innovative update strategy is proposed based on the Ebbinghaus Forgetting Curve in updating stage of UUBP algorithm, which can effectively avoid the influence of user's infrequent operation records and slowly forget the out-of-date records like a human being.
- Although our algorithm has been deployed to a server by our partner company, and the practice has proved that our algorithm meets the needs of the current smart home industry and has good effectiveness in user behavior prediction, we still evaluated our algorithm using 3 real data sources consisting of 10 devices, justifying the advantages of our approach over 3 widely-used clustering algorithms.

The remainder of this paper is organized as follows: Section 2 outlines several related prediction methods and discusses their performance from differing perspectives. The basic framework and detailed derivation of the proposed algorithm are explained in Section 3. In Section 4, the details of the experiments are addressed, and the experimental results demonstrate the performance of the UUBP algorithm is superior to traditional and existing clustering algorithms. Finally, the conclusion is stated within the final section of this paper.

## II. RELATED WORKS

In order to offer a personalized service and user experience, the historical and real-time operation data is collected by sensor devices in daily life. Subsequently, many algorithms are employed to mine the hidden valuable rules, and further predict the user's behaviors. For example: on weekdays, a user usually wakes up at 9AM and then operates the toaster. A superior user behaviors prediction method should mine this behavior of this user within the operation records and return this behavior to the smart home control center. The system will ask the user one day in advance whether needs to assist them in utilizing the toaster at 9AM tomorrow, if tomorrow is a weekday. Recently, there have many methods has tried to do this task and homes intelligently. The previous methods can be divided into two categories according to whether there is a need to initialize the amount of user behaviors.

Pingfan [15] puts forward a prediction method based on K-means and Particle Swarm Optimization (PSO) algorithm. The model also simulated the controlling of the electric curtains of the smart home and demonstrated that it can improve the learning ability of the home control system for user behaviors. Similarly, K-means algorithm has been employed to separate the normal routines from the suspected routines in order to monitor the change in the daily routine of a person living in a smart home and learn the user behaviors, where daily routine is the group of activities [16]. Those algorithms have tried to predict $k$ behaviors of a user by partitioning $n$ user operation records into $k$ clusters, but they need upon clustering number $k$. On many occasions, most user operation records are dynamic, and it is not known in advance how many user behaviors categories a given data set should require. Furthermore, the clustering number always has been given according to the past experiences, and the subjective experiences may ultimately influence the performance of the prediction algorithm.

Therefore, some methods that do not need manual intervention during the initialization stage have been proposed. For instance, Kim *et al.* [17] attempt to design a Recommendation Agent System (RAS) which can learn user behavior and provide recommendation service by considering the circumstance and desire of users using Hidden Markov Model (HMM)-based Collaborative Filtering (HCF). However, one major drawback of this method is that while the HMM algorithm is suitable for making predictions on a small data set with a single marked feature, they cannot be used within a big data scenario. Fatima *et al.* [18] recognize the daily life activities and predict user behavior by using a decision fusion of four individual Support Vector Machine (SVM) kernel functions, where each kernel is designed to learn the performed activities in parallel. But SVM is not suitable for multi-classification problems. Furthermore, as for high-dimensional data, it is hard to determine a reasonable kernel function for SVM. Additionally, some researchers have tried to use deep learning algorithm to perform this prediction task. For example, the Deep Belief Network-reconstruct algorithm (DBN-R) based on the deep learning framework for predicting user behaviors in a smart home has been proposed [19]. This method can solve the problem that requires the initialization for the amount of user behaviors exist in K-means. Yet, regrettably, the DBN-R algorithm demonstrates an accuracy of 43.9% (51.8%) for predicting newly activated sensors (smart home devices) based on MIT home data set 1 (data set 2). To summarize, those methods can better adapt to dynamic user data and do not need artificial participation to set the clustering number during the initialization stage of user behaviors prediction and clustering, but they still possess serious defects in their overall performance. In their view, each user operation record is equally important and has the same influence in the prediction of user behaviors. However, user behavior will change dynamically over a period

of time. Therefore, an excellent prediction algorithm should distinguish which user operation records are more important and try to forget the out-of-date records.

Therefore, this research aims to solve the above problems which exist in the previous prediction algorithms. The improved and novel unsupervised user behaviors prediction algorithm (UUBP) is proposed as a potential solution to the problem.

## III. THE PROPOSED ALGORITHM

In the proposed UUBP algorithm (ALGORITHM I), the proposed ANN is used to initialize the UUBP algorithm instead of requiring manual intervention. Also, in the updating stage of the UUBP algorithm, an innovative update strategy which has a forgetting factor based on the Ebbinghaus Forgetting Curve is proposed to remove the influence of user's infrequent and out-of-date operation records according to their respective generation date. This can reduce the impact of the out-of-date records during the prediction, in order to generate predictive behaviors which are closer to the recent user behaviors.

The proposed prediction algorithm is composed of five stages: date preprocessing, initialization stage, assignment stage, update stage, and user behavior generation:

a) Date preprocessing: Data preprocessing is an important step in the data mining process. If there is much difference in the format and missing-feature data points, then knowledge discovery during the training phase is increasingly difficult. Therefore, within the data preprocessing stage of UUBP algorithm, data transformation and data cleansing will be completed to avoid this problem.

b) Initialization stage: In UUBP algorithm, the user's operation records of a certain smart home device will be input into an ANN to execute the initialization stage in order to get the number of clusters and the respective centroid vector of each cluster automatically without manual setting.

c) Assignment stage: Assign each data point to the cluster whose centroid vector has the least squared Euclidean distance to this record according to the operation time and the operation state of the smart home device.

d) Update stage: In the UUBP algorithm, a forgetting factor will be integrated to propose a novel update strategy and calculate the new centroid vectors of each cluster in the new clusters.

e) User behavior generation: The final centroid vectors have to be transformed to a format which can be comprehended by the user.

The algorithm has converged when the assignments no longer change, and the final centroids are the user's predicted behaviors of this smart home device.

### A. DATA PREPROCESSING

Our data set is offered by a real in-situ smart home company and the company's experts have pointed out that the generation date of record, the operation time of device, and the operation state of device are the most important features in this prediction task. Therefore, there is no need to use algorithms to complete the features selection. Therefore, within the proposed UUBP algorithm, data preprocessing is conducted within two parts: data transformation and data cleansing.

a) Data Transformation: Data transformation allows the mapping of the data from its given format into the format expected by the prediction algorithm. In the UUBP algorithm, the generation date of the record, the duration of this activity, and the operation state of device will be mapped to the same format by using Equation (1), (2), and (3):

$$date(i)' = 10\frac{date(i) - \min(date)}{\max(date) - \min(date)} \tag{1}$$

Where $date(i) = t_i - t_0$, $t_i$ means the generation date of the $i$th record and $t_0$ means the run-date of the UUBP algorithm.

$$time(i)' = 10\frac{time(i) - \min(time)}{\max(time) - \min(time)} \tag{2}$$

Where $time(i) = 60(m_i + 60h_i)$, $h_i$ means the hours in operation time of the $i$th record while $m_i$ means minutes.

$$state(i)' = 10\frac{state(i) - \min(state)}{\max(state) - \min(state)} \tag{3}$$

In the data set, the operation state of a device is formatted by an integer, in binary form. For example, '0' indicates to turn off this device while '1' indicates to turn on. $state(i)$ means the operation state of the $i$th record.

b) Data Cleansing: the main task of the data cleaning part is missing date processing. In the UUBP algorithm, the Newton polynomial is used to perform the missing date interpolation.

### B. INITIALIZATION STAGE

In the standard K-means algorithm or other K-means-based algorithms, the initialization stage is to set the clustering number $k$ manually and utilize an initialization method to initialize $k$ centroid vectors. A commonly used initialization method is Random Partition [20]. However, these types of initialization method have a serious defect. If the randomly initial centroid vectors seriously deviate from the potentially current centroids, the algorithm will make many iterations to update the centroid vectors and therefore will be very inefficient. Furthermore, the clustering number which has been given according to the past and subjective experiences may ultimately influence the performance of the prediction algorithm. In this context, an ANN will perform the initialization task and as the initial coarse clustering to output $k$ centroid vectors which are closer to the potentially current centroids compared with the previous initialization method.

The ANN which is shown in Figure 2 has five layers, an input layer, a hidden layer, a reconstruction layer and two competitive layers. The input layer is mainly responsible for the input sample. The hidden layer tries to compress the input sample into a short code, and then the reconstruction layer will decompress that code into something that closely matches the original input data by using Equation (4).

**Algorithm 1** *UUBP* (n, dataSet)

**Input: n:** the number of rings in the equal probability models; all equal probability models use the same value of n.
**dataSet:** user's operation records for a certain smart home device.
**Output:** the predictive user behaviors for this device.
**1.** date preprocessing: transform all data in the dataSet to the same format and complete the data cleaning task.
**2.** initialization stage: input the dataset into the proposed ANN to attain $k$ clusters and the respective centroid vector of each cluster.
//assignment and updating stage
**3. while** the assignments change **or** never done assignment task:
**4. do**
**//** assignment stage
**5.** **if** never done assignment task **then**:
**6.** assign each data point to the cluster whose centroid vector has the least squared Euclidean distance to it according to the centroid vectors generated by initialization Stage.
**7.** **else**:
**8.** assign each data point to the cluster whose centroid vector has the least squared Euclidean distance to it according to the centroid vectors generated by the previous updating.
**//** updating stage
**9.** a) build up a n-ring equal probability and get the forgetting factor of each data point.
**10.** b) combine the forgetting factors to calculate the new centroid vector of each cluster according to the new clustering result.
**11.** **end if**
**12. end while**
**13.** transform the final centroid vectors to the user behaviors of this smart home device.
**14. return** the predictive user behaviors.

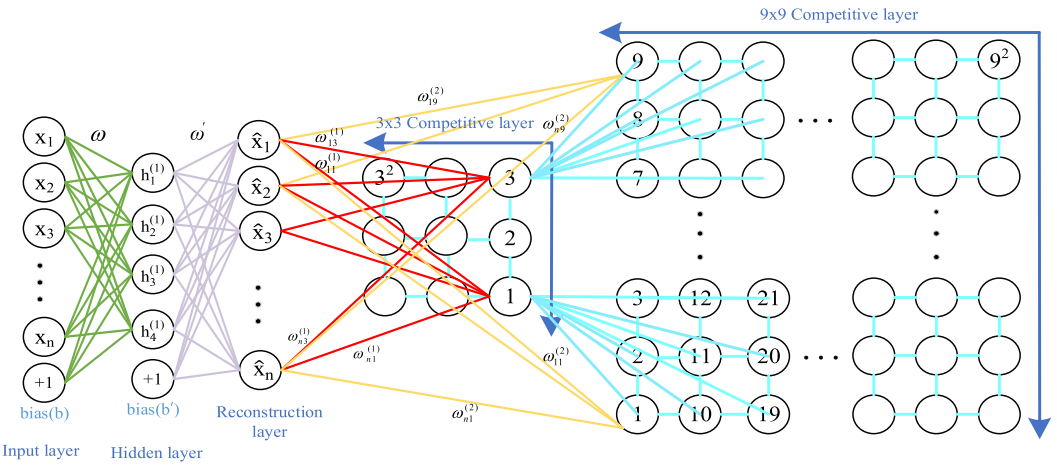


**FIGURE 2.** Framework of the proposed ANN.

Where $ReLU(x)$ means the rectified linear unit function [21] and $\sigma$ means the sigmoid function. $\omega, b$ mean the weights and bias between the input layer and the hidden layer while $\omega'$, $b'$ mean the weights and bias between the hidden layer and the reconstruction layer. During this procedure, the first three layers are trained to minimize the reconstruction error. This forces the neural network to engage in dimensionality reduction like an auto-encoder dose [22]. In our context, root-mean-squared error (RMSE) is used as the reconstruction error which is shown as Equation (5), and if the RMSE value is less than 0.02 or the number of epochs is greater than 300, the reconstruction layer will output the $\hat{x}$ in the last epoch to the followed competitive layers. The neurons of the reconstruction layer are full-connected to the neurons of the two competitive layers by weights $\omega^{(1)}$ and $\omega^{(2)}$ respectively. And each neuron of the first 3*3 competitive layer is connected to the corresponding 3*3 neurons lattice in the second 9*9 competitive layer by using Equation (6). $n_i^{(1)} \rightarrow n_j^{(2)} \sim n_{j+2}^{(2)}$ means the $i$th neuron in the first competitive layer is connected to the $j$th to $(j+2)$th neurons in the second competitive layer, where $j$ can be calculated by using $i$ through Equation (7). And "/" means modulo

operation while "%" means complementation. For example, 1/3=0 and 1%3=1. Therefore, the 1st neuron in the first competitive layer is connected to the 1st, 2nd, 3rd, 10th, 11th, 12th, 19th, 20th, and 21st neurons in the second competitive layer which is shown in Figure 2.

$$\hat{x}_i = 10\sigma(\omega' \cdot ReLU\ (\omega x_i + b) + b') \tag{4}$$

$$L\left(x_i, \hat{x}_i\right) = RMSE\left(x_i, \hat{x}_i\right) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{n}} \tag{5}$$

$$n_i^{(1)} \to n_j^{(2)} \sim n_{j+2}^{(2)} \tag{6}$$

$$j = 27\left((i-1)/3\right) + 3\left((i-1)\%3\right) + 1 + 9z,$$
$$z = 0 \sim 2 \tag{7}$$

As for the competitive layer, the core layer of the proposed ANN, its main task is to perform the dot product of the output of the reconstruction layer in order to do the competitive learning (ALGORITHM II). Then, the neuron with the maximum value called Winner Neuron will win the competition and attain the right to update the weights of all the neurons in its adjacent domain, so that the neurons have a stronger response to the similar input by using Equation (8) [23]:

$$\omega\,(t+1) = \omega\,(t) + \eta(t) \cdot (x - \omega(t)) \tag{8}$$

where $\eta$ indicates the function of the training time ($t$) and the topological distance of the adjacent domain of the winner neuron ($n$) from the winning neurons which can be written as Equation (9) [24]:

$$\eta\,(t+1) = \eta(t) \cdot e^{(-n)} \tag{9}$$

### C. ASSIGNMENT STAGE

Data point assignment is an important stage and the discriminator that determines whether the learning process should be terminated within a clustering algorithm. In this stage, the UUBP algorithm will assign each data point (user operation record) to the cluster whose centroid vector has the least squared Euclidean distance to this record. This is according to the operation time and the operation state of the smart home device in this record by using Equation (10).

$$C_i^{(t)} = \{r_p : ||r_p - x_i^{(t)}||^2 \le ||r_p - x_j^{(t)}||^2 \forall_j, 1 \le j \le K\} \tag{10}$$

Here, $r_p$ indicates a data point which has two eigenvalues: the operating time and the state of the smart home device in this user operation record. $x_j^{(t)}$ indicates the centroid vector of $j$th cluster and $x_i^{(t)}$ indicates the centroid vector of cluster $C_i^{(t)}$ in the $t$th learning iteration and it also has two eigenvalues likes $r_p$. The Equation (10) means that $r_p$ is assigned to exactly cluster $C_i^{(t)}$.

**Algorithm 2** *CompetitiveLearning*(dataSet)

**Input:** dataSet: the output of the reconstruction layer
**Output:** clustering result ($k$ centroid vectors)
**1.** randomize the node weight vectors in a map
**2.** randomly pick an input vector $x$
**3.** initialize $\eta = 0.6$
**4. while** $\eta \ge 1e^{-6}$ **:**
**5. do**
**6.** **while** do not traverse all node in the map:
**7.** **do**
**8.** 1) calculate the Euclidean distance between the input vector $x$and each node weight vector in the map
**9.** 2) track the node that produces the smallest distance (n) in the map and mark the node as the best matching unit (BMU)
**10.** **end while**
**11.** update the weight vectors of the nodes about BMU and BMU itself by pulling them closer to the input vector:
$\omega\,(t+1) = \omega\,(t) + \eta(t) \cdot (x - \omega(t))$
**12.** update $\eta$: $\eta\,(t+1) = \eta(t) \cdot e^{(-n)}$
**13. end while**
**14. return** clustering result

### D. UPDATING STAGE

In the traditional clustering algorithm, the strategy of updating the centroid vector is susceptible to the bias created by outliers and does not take the temporal feature of the user records into consideration well. Therefore, it will not be able to effectively identify which record is more important and learn doing forgetting learning like a human being. However, inspired by the Ebbinghaus Forgetting Curve [25], in this paper, there is a novel the strategy of updating the centroid vector to the updating stage by adding a forgetting factor in the UUBP algorithm. Here, it is demonstrated that this algorithm, which uses a new strategy, is able to address some of the problems which currently exist in traditional algorithms. This allows for the easy identification of outliers which are indicated by the infrequent and out-of-date operation records of the user which may be far from the most recent user behaviors.

#### 1) THE STRATEGY FOR CENTROID VECTOR UPDATE

The improved updating strategy (ALGORITHM III) will use the forgetting factor $\omega(r_i)$ as an important index to update the centroid vector. This is in order to remove the influence of the user's infrequent operation records and mine the user's commonly recent control behaviors effectively. It can be written as the Equations (11) to represent the strategy for updating the centroid vector of our improved K-means algorithm. Where $r_i$ indicates a data point in this certain cluster.

$$\overline{D} = \frac{\sum_{i=1}^{n} \omega(r_i) \cdot r_i}{n} \tag{11}$$

**Algorithm 3** *UpdatingStrategy* (assignment, centroids, n)

**Input:** assignment: assignment of data points ($k$ clusters)
centroids: $k$ centroid vectors of $k$ clusters
n: the number of rings in the equal probability model

**Output:** $k$ new centroid vectors

**1.** build a n-ring equal probability model
**2. for** $(i = 0; i++; i < k)$:
**3. do**
**4.** calculate the forgetting factors $\omega(r_i)$ for each data points $r_i$ in the $i$th cluster.
**5.** update the centroid vector of $i$th cluster:

$$\overline{D} = \frac{\sum_{i=1}^{n} \omega(r_i) \cdot r_i}{n}$$

**6. end for**
**7. return** $k$ new centroid vectors.

### 2) THE FORGETTING FACTOR

The Ebbinghaus Forgetting Curve hypothesizes that the process of forgetting is non-linear. The initial process of forgetting is fast and then slows down. Therefore, the concept would mean that the more previous knowledge is easier to be forgotten [26]. Similarly, user behaviors will change over time. For example, a user was a worker two years ago and the individual had to get up at 7am every workday, and then work their toaster to serve breakfast. But, a change of circumstance such as retirement, would mean their wake-up time has changed to 9 am and then use the toaster. Thus, as an excellent user behaviors prediction algorithm, it should reduce the weight of this user operation records generated around two years ago while increasing the weight of those records generated recently during the predictive learning. The algorithm should realize which records are more important and which should be forgotten. Thus, under the influence of the Ebbinghaus Forgetting Curve, it is considered that the learner should gradually forget the user's operation records according to its generation date, as like a human would do in order to mine behaviors which are closer to user's recent behaviors. So, it is proposed a forgetting factor model to complete this task is defined in Equation (12):

$$\omega(r_i) = \exp(-\frac{date(r_i)'}{p_i}) \tag{12}$$

Here, $r_i$ indicates a certain data point in this cluster which is a user operation record. $p_i$ indicates the probability parameters of this record $r_i$, it is proposed to promote the convergence of the clustering process and magnitude the difference between each record to improve the importance of generation date during the prediction clustering, and we will introduce it next in next section. $date(r_i)'$ can be calculated by using Equation (1). From Equation (12), we can see that each record will be given a weight according to its generation date, and the more historical record always has a smaller forgetting factor weight.

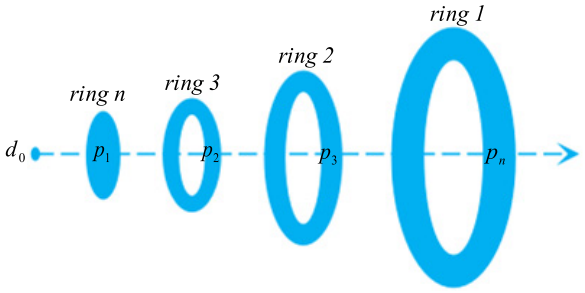


**FIGURE 3.** The equal probability models.

### 3) THE PROBABILITY PARAMETER

From Equation (12), it is recognized that each data point possesses a probability parameter which is proposed to promote the convergence of the clustering process and magnitude the difference between each record. This improves the importance of the generation date during the prediction clustering. The probability parameter which this research proposes is a weight factor based on its generation date $date(r_i)'$, and the value of the probability parameter decreases stage by stage. According to the definition of the arithmetic progression, we propose the Equal Probability Model which is shown in Figure 3.

In order to build the model as shown in Figure 3, for a certain cluster, data points of this cluster must be divided into $n$ regions based on the following procedure:

a) Find the maximum $date(r_i)'$ and denote as max?($date'$).

b) For every data point $r_i$, if there exists an integer $k$ between 0 to $n$ which can make its $date(r_i)'$ satisfy the Equation (13), then the data point $r_i$ will be assigned to the ring $k$.

$$\frac{\max(date')}{n}k < date(r_i)' \leq \frac{\max(date')}{n}(k+1) \tag{13}$$

Then, each ring will be assigned a different probability parameter ($p_i$) where the sum of all the $p_i$ is 1 when calculated based on the following procedure:

a) Calculate the probability parameters of the points in the outermost ring ($p_n$) using Equation (14):

$$p_n = \frac{2}{n(n+1)}, \quad n > 1 \tag{14}$$

b) Calculate the probability parameters for the points in the $i$th ring ($p_i$) using Equation (15):

$$p_i = (n - i + 1)p_n \tag{15}$$

Visualizing the equations for the Equal Probability Model of this cluster will result in a diagram as like the one shown in Figure 3. From its definition and Figure 3, it can conclude that the recent records which are closer to the current date will have a smaller probability parameter $p$ and a much bigger forgetting factor $\omega$ compared with the antiquated records. This strategy can magnitude the difference between the two types of records and improve the importance of generation date.

**TABLE 1.** Data set.

| Data Set | # records | # devices |
|---|---|---|
| Research and Development Center (ID: 1335) | 17714 | 3 |
| Program Test Room (ID: 1660) | 17455 | 3 |
| Exhibition Hall (ID: 1696) | 22718 | 4 |
| **Total** | 57887 | 10 |

### E. USER BEHAVIOR GENERATION

The final centroid vectors mean the feature vectors which cannot be comprehended by user and are inconsistent with the default format. Therefore, the values of the final centroid vectors have to be mapped back to their original formats respectively. In UUBP algorithm, the predictive user behavior is composed by the operation time and the operation state of device. The operation time of device can be mapped to its original format by using Equation (16), (17), and (18) while the operation state uses Equation (19):

$$time(i) = \lfloor 10time(i)' \cdot (\max(time) - \min(time)) + \min(time) \rfloor \quad (16)$$

$$Rh_i = time(i)/60 \quad (17)$$

$$Rm_i = time(i)\%60 \quad (18)$$

$$state(i) = \lfloor 10state(i)' \cdot (\max(state) - \min(state)) + \min(state) + 0.5 \rfloor \quad (19)$$

Where $\lfloor x \rfloor$ means the function that rounding down $x$. For example, $\lfloor 12.59 \rfloor = 12$. "/" means modulo operation while "%" means complementation. $Rh_i$ means the recommend operation hour while $Rm_i$ means operation minute.

## IV. EXPERIMENTS AND ANALYSIS

In this section, the experiments are performed to verify the effectiveness of the proposed UUBP algorithm based on 3 real data sources consisting of 10 devices from a smart home company and each data set is directly linked to a smart home which is shown in TABLE 1.

TABLE 2-4 list the devices for each data set. It must be acknowledged that the removal of all records containing missing values has occurred. Also, the utilization of the generation date of the record, the generation time of this operation activity, and the operation state of device as important features of user behaviors clustering for a certain device is noted by company engineers.

The platform for calculation is a personal computer with an Intel(R) Core(TM) i7-6770 3.40 GHz CPU, and 8 GB random-access memory, running Windows 7 Professional operating system. All algorithms were coded by Python programming language.

In order to explain the superiority of UUBP algorithm, the use of some classic and widely-used algorithms such as K-means, SOMNN and DBSCAN algorithm to undertake the same tasks as comparative experiments.

The Compactness Index(CP), Separation Index(SP) and the Davies-Bouldin index (DB) are always used to evaluate the effectiveness of a clustering algorithm [27]. There is a

**TABLE 2.** Devices in 1335.

| Data Set | 1335 | |
|---|---|---|
| attributes | # device name | # records |
| | Smart Curtain (SC) | 5911 |
| | Headlight (HL) | 5221 |
| | Exhaust Fan (EF) | 6582 |
| Total | 3 | 17714 |

**TABLE 3.** Devices in 1660.

| Data Set | 1660 | |
|---|---|---|
| attributes | # device name | # records |
| | Air Conditioning (AC) | 6182 |
| | Smart Light (SL) | 4798 |
| | Smart Switch (SS) | 6475 |
| Total | 3 | 17455 |

**TABLE 4.** Devices in 1696.

| Data Set | 1696 | |
|---|---|---|
| attributes | # device name | # records |
| | Air Conditioning (AC) | 6943 |
| | Headlight (HL) | 5973 |
| | Smart Socket (SSK) | 5053 |
| | Curtain(CT) | 4749 |
| Total | 4 | 22718 |

modification of the equation for calculating CP by adding a Time-Distance factor (TD) to measure whether the algorithm can effectively distinguish the out-of-date operation records and effectively locate the user's recent control behaviors which can be calculated by using Equation (20).

$$TD_i = |t_{\omega_i}| \quad (20)$$

Where $t_{\omega_i}$ indicates the average generation date of all records $average(date(x)')$ in ith cluster that is the generation date of its centroid, and $t_{r_i}$ means the generation date of the first recent record in ith cluster. In this context, the final clustering centroids are the predictive user behaviors. Therefore, the bigger TD the algorithm, the worse recognition and forgetting ability of the out-of-date operation records it has.

The CP value calculates the average distance from each point to its centroid respectively. The smaller the value of CP, the more compact the data is within the cluster. The CP value can be calculated by using Equation (21) and (22):

$$\overline{CP_i} = \frac{1}{|\Omega_i|} \sum_{x_i \in \Omega_i} ||x_i - \omega_i|| \quad (21)$$

$$CP = \frac{1}{k} \sum_{i=1}^{k} ||\overline{CP_i} + TD_i|| \quad (22)$$

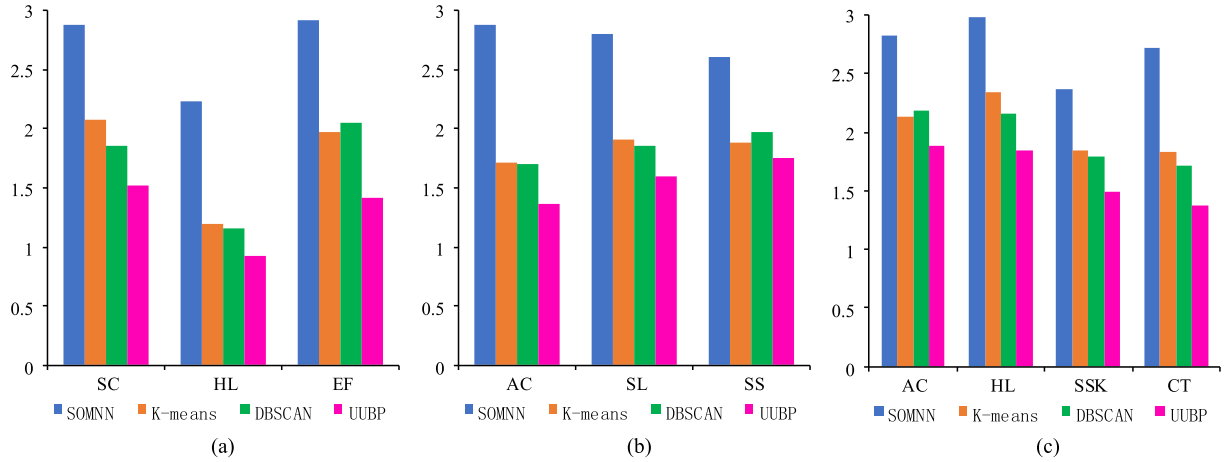


**FIGURE 4.** The average CP of each device. (a) Devices in 1335. (b) Devices in 1660. (c) Devices in 1669.

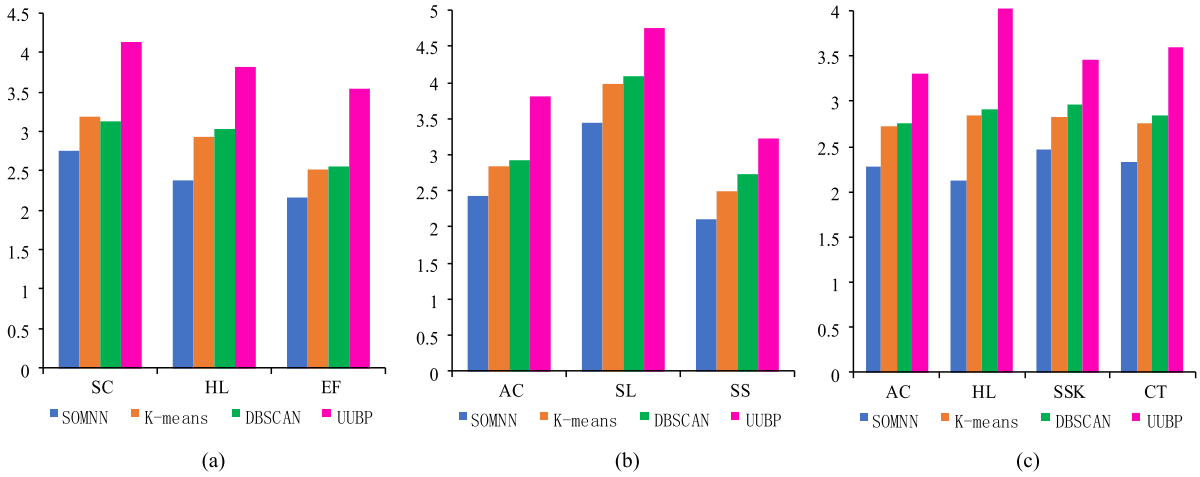


**FIGURE 5.** The average SP of each device. (a) Devices in 1335. (b) Devices in 1660. (c) Devices in 1669.

The *SP* value calculates the average distance between two different centroids which can be calculated by using Equation (23) and the higher *SP* means looser between two clusters.

$$SP = \frac{1}{k^2 - k} \sum_{i=1}^{k} \sum_{j=i+1}^{k} ||\omega_i - \omega_i||_2 \tag{23}$$

The *DB* value comprehensively considers the *CP* value and the *SP* value. Therefore, *DB* value is generally used to evaluate the clustering effectiveness of a clustering algorithm [28]. This *DB* value can be calculated by using Equation (24).

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left( \frac{||CP_i - CP_j||}{||\omega_i - \omega_j||_2} \right) \tag{24}$$

In order to verify the advantage of the proposed UUBP algorithm in effectiveness, 5 repeated experiments have been performed by using each different smart home device respectively, totally 50 repeated smart-home-device experiments for each algorithm except K-means algorithm. As for K-means algorithm, make $k = 3 \sim 8$ and use the average CP, SP, and DB after 5 repeated K-value experiments for each $k$ value respectively as the *CP*, *SP*, and *DB* of K-means algorithm for each device, and each K-value experiment has 50 repeated smart-home-device experiments as mentioned above like other algorithms. This totals 300 K-means experiments which have been performed. Therefore, totally 450 experiments have been conducted for this research.

As for the experiments result, Figure 4 shows the average *CP* value of each algorithm for each device after the repeated experiments in the 3 data sets respectively while Figure 5 shows the average *SP* value and Figure 6 shows the average *DB* value.

As can be seen in Figure 4, the proposed UUBP algorithm has the minimum *CP* value, that is, the UUBP algorithm can put the similar user records together better while taking into account the generation date. And in Figure 5, it can be seen that the UUBP algorithm has the maximum *SP* value. So, it proves that the UUBP algorithm performed the best in separating different and dissimilar user records. Finally, as can

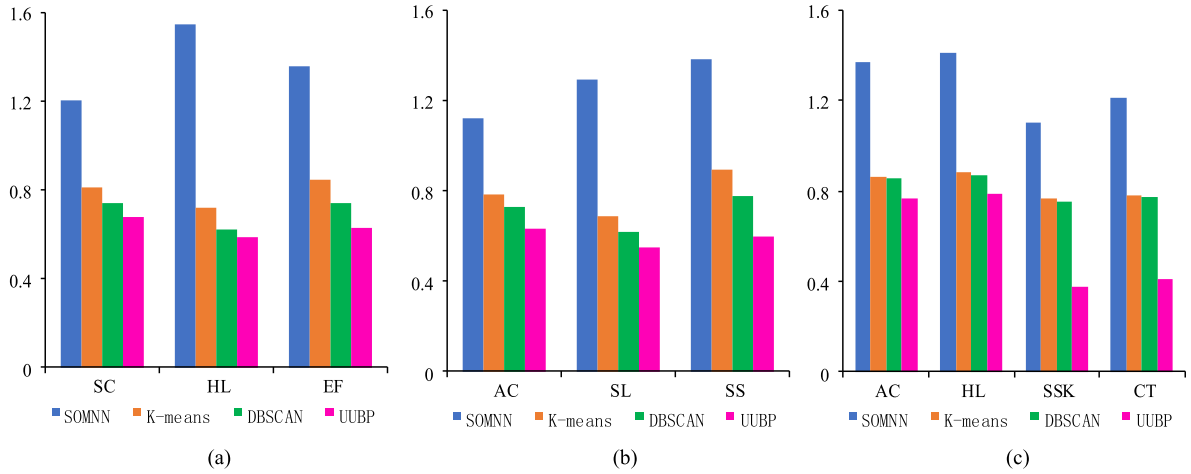


**FIGURE 6.** The average DB of each device. (a) Devices in 1335. (b) Devices in 1660. (c) Devices in 1669.

be seen in Figure 6, the *DB* value of the UUBP algorithm is the lowest one. Therefore, it can be concluded that the UUBP algorithm can better handle the out-of-date operation records and has the best performance in user behavior prediction combines temporal feature. Therefore, the UUBP algorithm has the best ability of recognizing and forgetting the out-of-date records, and its predictive user behaviors are closer to the users' recently real behaviors.

## V. CONCLUSION AND FUTURE WORK

One of the biggest challenges faced by a smart home system is how to successfully mine the potential value of the user operation records, and concurrently try to be a confidant of the user. So, user behaviors prediction is still a valuable and challenging area for research. This paper aims to tackle this challenge and overcome the shortcomings of the previous user behaviors prediction algorithms to improve the current state of the prediction task. The proposition of a novel user behaviors prediction algorithm based on machine learning and mathematical knowledge, namely the UUBP algorithm. In this improved algorithm, a forgetting factor model based on the equal probability model and the Ebbinghaus forgetting curve is integrated in order to remove the influence of out-of-date records and attain a much more satisfactorily predictive behavior. Additionally, in order to let the learner, have a strong autonomous learning ability, a novel ANN is used to helps initialize the learner. Finally, the experimental results show that compared to the previous prediction algorithms, the UUBP algorithm is much more excellent.

This paper is mainly aimed at how to efficiently mine user behaviors of a certain smart home device from user operation records. However, how to mine user's associative behaviors for a series of activities where a number of different devices are involved is equally important within smart home systems. For example, when the user wakes up, they turn on the coffee maker, make some toast, and go for a shower. The algorithm can predict which device will be manipulated after those activities are performed. Therefore, the main task of future research is to investigate a method which can predict the users associative controlling behaviors and can investigate how the method identify the temporal association among the activities, involving a variety of devices, their temporal characteristic, and mine user's associative behaviors.

## REFERENCES

[1] J. Wan *et al.*, "A manufacturing big data solution for active preventive maintenance," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2039–2047, Aug. 2017.

[2] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 94–100, Jul. 2017.

[3] J. Liu, J. Wan, Q. Wang, P. Deng, K. Zhou, and Y. Qiao, "A survey on position-based routing for vehicular ad hoc networks," *Telecommun. Syst.*, vol. 62, no. 1, pp. 15–30, 2016.

[4] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *J. Cleaner Prod.*, vol. 140, no. 3, pp. 1454–1464, 2017.

[5] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012.

[6] W. K. Edwards, R. E. Grinter, R. Mahajan, and D. Wetherall, "Advancing the state of home networking," *Commun. ACM*, vol. 54, no. 6, pp. 62–71, Jun. 2011.

[7] A. J. Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu, and C. Dixon, "Home automation in the wild: Challenges and opportunities," in *Proc. ACM SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 2115–2124.

[8] I. A. Berg, O. E. Khorev, A. I. Matvevnina, and A. V. Prisjazhnyj, "Machine learning in smart home control systems—Algorithms and new opportunities," in *Proc. AIP Conf.*, vol. 1906, no. 1, 2017, p. 070007.

[9] M. Z. Uddin and M. R. Kim, "A deep learning-based gait posture recognition from depth information for smart home applications," in *Advances in Computer Science and Ubiquitous Computing*. Singapore: Springer, 2016, pp. 407–413.

[10] L. Wei, J. Bo, and C. Jie, "Research of the segmentation strategy of customer behavior in chinese mobile market based on the K-means arithmetic," *Chin. J. Manage.*, vol. 2, no. 1, pp. 80–84, 2005.

[11] M. Chan, E. Campo, D. Estève, and J.-Y. Fourniols, "Smart homes—Current features and future perspectives," *Maturitas*, vol. 64, no. 2, pp. 90–97, 2009.

[12] K. S. Gayathri, K. S. Easwarakumar, and S. Elias, "Contextual pattern clustering for ontology based activity recognition in smart home," in *Proc. Int. Conf. Intell. Inf. Technol.* Singapore: Springer, 2017, pp. 209–223.

[13] A. K. Jain, "Data clustering: 50 years beyond *k*-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[14] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, "Incorporating temporal reasoning into activity recognition for smart home residents," in *Proc. AAAI Workshop Spatial Temporal Reasoning*, 2008, pp. 53–61.
[15] L. Wang and P. Shao, "Intelligent control in smart home based on adaptive neuro fuzzy inference system," in *Proc. IEEE Chin. Automat. Congr. (CAC)*, Nov. 2015, pp. 1154–1158.
[16] L. G. Fahad, A. Ali, and M. Rajarajan, "Long term analysis of daily activities in smart home," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2013, pp. 419–424.
[17] J.-H. Kim, K.-Y. Chung, J.-K. Ryu, K.-W. Rim, and J.-H. Lee, "A recommendation agent system using HMM-based collaborative filtering in smart home environment," in *Proc. IEEE Converg. Hybrid Inf. Technol.*, vol. 2, Nov. 2008, pp. 214–217.
[18] I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee, "A unified framework for activity recognition-based behavior analysis and action prediction in smart homes," *Sensors*, vol. 13, no. 2, pp. 2682–2699, 2013.
[19] S. Choi, E. Kim, and S. Oh, "Human behavior prediction for smart homes using deep learning," in *Proc. IEEE Int. Alsymposium Robot Hum. Interact. Commun.*, Aug. 2013, pp. 173–179.
[20] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proc. ACM 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 600–607.
[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (IMLS)*, 2010, pp. 807–814.
[22] D. P. Kingma and M. Welling. (Dec. 2013). "Auto-encoding variational Bayes." [Online]. Available: https://arxiv.org/abs/1312.6114
[23] M. M. Mostafa, "Clustering the ecological footprint of nations using Kohonen's self-organizing maps," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 2747–2755, 2010.
[24] M. H. Ghaseminezhad and A. Karami, "A novel self-organizing map (SOM) neural network for discrete groups of data clustering," *Appl. Soft Comput.*, vol. 11, no. 4, pp. 3771–3778, 2011.
[25] S. K. Carpenter, H. Pashler, J. T. Wixted, and E. Vul, "The effects of tests on learning and forgetting," *Memory Cognition*, vol. 36, no. 2, pp. 438–448, 2008.
[26] H. Yu and Z. Li, "A collaborative filtering method based on the forgetting curve," in *Proc. IEEE Int. Conf. Web Inf. Syst. Mining (WISM)*, vol. 1, Oct. 2010, pp. 183–187.
[27] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002.
[28] J. C. R. Thomas, M. S. Peñas, and M. Mora, "New version of Davies-Bouldin index for clustering validation based on cylindrical distance," in *Proc. IEEE 32nd Int. Conf. Chilean Comput. Sci. Soc. (SCCC)*, Nov. 2013, pp. 49–53.

**TIANKAI LIANG** was born in Zhaoqing, Guangdong, China, in 1993. He received the B.S. degree in Internet of Things engineering from the Xi'an University of Science and Technology in 2016. He is currently pursuing the M.S. degree in computer engineering with the Guangdong University of Technology (GDUT), China.

His research interests include data mining, artificial intelligence, and machine learning. He received the First-Class Scholarship from GDUT, the National Inspirational Scholarship from the Ministry of Education of China, and the Third Prize in mathematics competition from the Shaanxi Provincial Department of Education.

**BI ZENG** received the M.S. and Ph.D. degrees from the Guangdong University of Technology (GDUT). She is currently a Professor with the School of Computers, GDUT. Her current research interests include computational intelligence, data mining, intelligent robot, and wireless sensor networks. She is a Senior Member of CCF, Multi-Valued Logic and Fuzzy Logic Committee, China.

**JIANQI LIU** (M'10) received the M.S. degree in computer software and theory from the School of Computer, Guangdong University of Technology (GDUT), in 2009, and the Ph.D. degree in control science and engineering from the School of Automation, GDUT, in 2016. He is currently an Associate Professor with the School of Automation, GDUT, China. His current research interests are big data technologies, Internet of Vehicle, and cyber-physical systems.

**LINFENG YE** received the B.S. degree from Xidian University, China, in 2002, the M.S. degree from Pierre and Marie Curie University, France, in 2007, and the Ph.D. degree in electrical and computer engineering from the Université de Bretagne Sud, Lorient, France, in 2011. He is currently an Assistant Professor in computer engineering with the Guangdong University of Technology. His work focuses on self-adaptive reconfigurable computing, artificial intelligence, and computer vision.

**CAIFENG ZOU** received the B.S. degree from Shanghai University, Shanghai, in 2006, the M.S. degree from Sun Yat-sen University, Guangzhou, China, in 2008, and the Ph.D. degree from the South China University of Technology, Guangzhou, in 2017. She is currently an Associate Professor with the Guangdong Mechanical and Electrical College, Guangzhou. Her current research interests include data mining, big data, cloud computing, and Internet of Things.