

Article

Recognition of Daily Activities of Two Residents in a Smart Home Based on Time Clustering

Jinghuan Guo ^{1,2,*}, Yiming Li ¹, Mengnan Hou ¹, Shuo Han ¹ and Jianxun Ren ¹

¹ School of Information Science & Technology, Dalian Maritime University, Dalian 116026, China

² Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong 643000, China

* Correspondence: guojinghuan@dlnu.edu.cn

Received: 5 January 2020; Accepted: 27 February 2020; Published: 6 March 2020



Abstract: With the development of population aging, the recognition of elderly activity in smart homes has received increasing attention. In recent years, single-resident activity recognition based on smart homes has made great progress. However, few researchers have focused on multi-resident activity recognition. In this paper, we propose a method to recognize two-resident activities based on time clustering. First, to use a de-noising method to extract the feature of the dataset. Second, to cluster the dataset based on the begin time and end time. Finally, to complete activity recognition using a similarity matching method. To test the performance of the method, we used two two-resident datasets provided by Center for Advanced Studies in Adaptive Systems (CASAS). We evaluated our method by comparing it with some common classifiers. The results show that our method has certain improvements in the accuracy, recall, precision, and F-Measure. At the end of the paper, we explain the parameter selection and summarize our method.

Keywords: activity recognition; sensor; smart home

1. Introduction

The problem of aging in the world's population is becoming increasingly serious; meaning, the proportion of the aging population is increasing while the fertility rate continues to decrease. By the end of 2019, the number of elderly people worldwide exceeded infants and young children. The problems brought about by an aging population are not only political and financial; in the reality of human aging, brain power will gradually weaken, and symptoms, such as decreased memory and brain function, will occur. The elderly care problem brought about by this is becoming increasingly prominent. Some empty-nest seniors are old and frail, and there is no one to take care of them in emergencies, such as falls, which may cause irreparable losses.

To improve this problem, some researchers have begun to focus on elderly activity recognition in smart homes. Elderly activity recognition is an important part of the functioning of smart homes, mainly as it could determine abnormalities in the elderly by obtaining information about their daily activities. This can help predict some potential diseases in the elderly, such as Alzheimer's disease, which is also the main motivation for many activity recognition studies in intelligent environments (and has widely been recognized by families affected by Alzheimer's disease) [1].

According to different data collection methods, smart home activity recognition can be roughly divided into intrusive and non-intrusive design.

Intrusive design usually refers to video-based activity recognition. It mainly records the daily lives of the elderly through cameras, and then methods are used to analyze the video for activity recognition [2–4]. The effect of video-based activity recognition is excellent [5], but its disadvantages

are also very prominent. First, it collects a lot of privacy and sensitive information. Second, it is susceptible to light. Finally, it is very expensive.

For non-intrusive design, it is mainly divided into wearable and environmental interaction activity recognition. Wearable devices mainly refer to miniature sensors, such as accelerometers and gyroscopes [6–9], which can be worn on the body. When older people wear them, they can collect information anytime, anywhere. However, wearing these devices can be a burden and cause resentment in the wearers.

With the development of the Internet of things technology [10,11], smart homes have become more complete, making activity recognition popular. This method mainly interacts with seniors by embedding sensors in objects in the smart home. These sensors are mainly divided into environmental sensors and binary sensors. Environmental sensors can collect environmental information in real time. Binary sensors can interact with seniors to locate them. This method is secretive and convenient, and will not cause resentment in the elderly.

In this paper we focus on environment-based interactive activity recognition. Many commonly used methods characterized by statistical sensor frequency ignore the temporal correlation of features. In order to improve the methods, we propose a data-driven activity recognition method. The method is divided into three stages: feature extraction, temporal clustering, and activity recognition. The method of the feature extraction proposes a de-noising method to remove interference from other residents. Generally it is believed that the daily activity of the elderly has a certain regularity, so we can cluster the activities in the close time. In the end, the similarity matching equation proposed in the paper is employed to calculate out the points with the most similarity, and select the classification by the votes of the points.

The structure of this paper is as follows: Section 2 introduces related work. Section 3 defines some terminologies. Section 4 describes our method. In Section 5, we validate our method and discuss it. Section 6 concludes the paper.

2. Related Work

Activity recognition based on non-intrusive design can be divided into data-driven and knowledge-driven, according to the differences of the used methods.

The knowledge-driven method extracts things, space, and time into certain rules in the domain, constructs them into a reusable context model and correlates with activities, and then uses inference and other technologies to determine the activity category. Ontology is often used in knowledge-driven methods. Chen et al. [12] proposed a formal explicit ontological modeling and representation of the smart home domain approach to the processing of multisource sensor data streams. Ye et al. [13] proposed a hierarchical structure of the domain concept ontology model to represent domain knowledge, which is independent of particular sensor deployment and activities of interest. Their subsequent researches [14] combined the ontology and statistical methods to automatically detect the boundaries of different activities and extended the Pyramid Match Kernel (PMK) to accommodate and balance the sensor noise in activity recognition. Ontology is also used for cross-environment activity recognition. Wemlinger et al. [15] proposed a Semantic Cross-Environment Activity Recognition (SCEAR) system to establish different ontology models for 22 data sets provided by CASAS to map between different environments. This method has a good recognition effect on two smart homes with similar layouts. The knowledge-driven method can clearly distinguish activities with significant semantic differences, but cannot identify well the activities with similar semantics.

Compared with knowledge-driven, data-driven models place more emphasis on the use of large-scale data for reasoning to build decision models [16]. Data-driven is mainly focused on supervised methods to recognize unlabeled activities by generating classifiers with labeled activities. Ravi et al. [17] proposed a naive Bayesian approach to activity recognition using accelerometers. Ruben et al. [18] proposed a resident adaptation technique based on Hidden Markov Models (HMMs). This system segments and recognizes six different physical activities using inertial signals from a

smartphone. Asghari P. et al. [19] proposed an online hierarchical hidden Markov model to predict the activity in the environment with any sensor event. This method first uses the HMM model to recognize the beginning and end of an activity, and then predicts the next activity by establishing an HMM for each sensor event.

Some ensemble methods have also been applied in activity recognition. Hu et al. [20] proposed a novel separating axis theorem (SAT) based splitting strategy, then used it to improve the random forest. Anna et al. [21] used a previously developed Cluster-Based Classifier Ensemble [22] (CBCE) method for smart home-based activity recognition; this method proposes a support formula for clustering to solve the recognition problem in clusters. In addition, with the development of deep learning, neural networks have also been applied to activity recognition. Arifoglu D et al. [23] extracted fixed-length sliding windows into a sparse two-dimensional time matrix to use Convolutional Neural Networks (CNN) for activity recognition. This is basically the same as the work of Gochoo et al. [24], and both have been tested on the Aruba dataset (and the performance is roughly the same). However, Arifoglu et al. proposed a novel method for identifying abnormal activities, and discovered abnormalities, as well as identified and prevented abnormal activities by generating abnormal activities in the data set. Medina et al. [25] proposed a method using fuzzy time windows (FTW) to segment the data set, followed by Long Short-Term Memory (LSTM) for activity recognition.

There are also unsupervised and semi-supervised methods. The semi-supervised method mainly uses a small part of the labeled data to label a large amount of data; thereby, reducing the workload of the labeling activity. Hu et al. [26] proposed a cross-domain activity recognition (CDAR) algorithm to label another set of different but related activity from labeled activities. Wen [27] et al. proposed a similarity measurement formula that uses a small amount of labeled data to label a large amount of unlabeled data. The difficulty of the unsupervised method is the problem of data labeling. Researchers have now proposed some unsupervised methods to solve the problem of data annotation, such as frequent sensor mining methods [28], and frequent periodic pattern mining methods [29], activity modeling based on low-dimensional feature space [30], probabilistic model [31,32], and retrieval of activity definition, using Web mining [33].

Some multi-resident activity recognition based on smart homes has also been proposed. Hao et al. [34] proposed a knowledge-driven solution based on formal concept analysis (FCA) and sequential pattern mining to analyze the activity rules of different residents and recognize human activities in non-intrusive sensor data. Alemdar et al. [35] used the factorial hidden Markov model and nonlinear Bayesian tracking to recognize the behaviors of the two residents. They did not distinguish between the residents and achieved good results. Guo et al. [36] proposed a method to extract features using the improved Term Frequency-Inverse Document Frequency (TF-IDF). They used the improved TF-IDF to calculate the probability of the sensor appearing in the activity, recognized it as a new feature, and tested the method on the Tulum2009 and Cairo datasets. Lu et al. [37] used Back Propagation-Hidden Markov Model (BP-HMM) to extract daily activity features and SVM to recognize daily activity. They introduced the dependent Beta process into the HMM, and integrated the state constraints of the sensors into the sampling process. Finally, SVM is employed to recognize daily activities.

Many multi-resident activity recognition focuses on feature extraction. This paper proposes a clustering method based on the time pattern of elderly activity, then based on this, a similarity matching formula is proposed. This formula is based on Levenshtein Distance which is used in the stage of activity recognition model development, rather than the stage of daily activity feature extraction. We applied this method to the activity recognition of two residents and the results were very attractive. Although the cost of training will be higher, this method is better than a large number of single and ensemble methods.

3. Terminologies

To better express our method, we define some terminologies and use the activities in Table 1 as examples.

Definition 1. $se = (d, t, s, ss, ar, as)$ is a sensor event, where d is the date when the event occurred, t is the time when the event occurred, s is the activated sensor, ss is the sensor status, ar is the corresponding activity, and as is the activity status.

In this paper we use $se.d, se.t, se.s, se.ss, se.ar, se.as$ to represent d, t, s, ss, ar, as in this event. Table 1 shows the two activities in the dataset and the meaning of $s = (d, t, s, ss, ar, as)$. For example, $se_1 = (d, t, s, ss, ar, as)$ represents $se_1 = (2010-11-04, 05:40:51.303739, M004, ON, Bed_to_Toilet, begin)$.

Definition 2. For a given sensor event $se_1, se_2 \dots se_n$, we connect their s in series, $sq = \{se_1.s, se_2.s \dots se_n.s\}$ is called a sensor sequence, which is simplified to $sq = \{s_1, s_2 \dots s_n\}$.

The sensor sequence of Bed_to_Toilet activity in Table 1 is expressed as $sq_1 = \{M004, M005, M007, M001, M004, M004, M007, M001\}$, and the sensor sequence of Sleep is expressed as $sq_2 = \{M004, M004.M007, M007, M006, M007, M005, M004\}$. Note that we use the part where the activities intersect as common to both activities.

Table 1. A segment of activity records.

id	d	t	s	ss	ar	as
se ₁	2010-11-04	05:40:51.303739	M004	ON	Bed_to_Toilet	begin
se ₂	2010-11-04	05:40:52.342105	M005	OFF		
se ₃	2010-11-04	05:40:57.176409	M007	OFF		
se ₄	2010-11-04	05:40:57.941486	M001	OFF		
se ₅	2010-11-04	05:43:24.021475	M004	ON	Sleep	begin
se ₆	2010-11-04	05:43:26.273181	M004	OFF		
se ₇	2010-11-04	05:43:26.345503	M007	ON		
se ₈	2010-11-04	05:43:26.793102	M007	ON	Bed_to_Toilet	end
se ₉	2010-11-04	05:43:27.195347	M006	OFF		
se ₁₀	2010-11-04	05:43:27.787437	M007	ON		
se ₁₁	2010-11-04	05:43:29.711796	M005	ON		
se ₁₂	2010-11-04	05:43:30.279021	M004	OFF	Sleep	end

Definition 3. For activity a and a series of sensor events, we represent the activity as $a = \{bt, et, sq\}$, where bt is $se_1.t$ retention hour, et is $se_n.t$ retention hour, and sq is the sensor sequence in Definition 2.

For example, the Bed_to_Toilet activity in Table 1 can represent $a = \{05, 05, sq_1\}$, where $sq_1 = \{M004, M005, M007, M001, M004, M004, M007, M001\}$.

Definition 4. We count the importance of other sensors in the activities that begin with a sensor and store them in the SIA. The algorithm is shown in Algorithm 1.

Algorithm 1: GettingSIA

Input: $Sq\{sq_1, sq_2, \dots, sq_n\}$, a set of sensor sequence
Sensor, sensor set

Output: *SIA*, two-dimensional tuple of sensor importance

1. $SIA \leftarrow \emptyset$;
2. $fre \leftarrow \emptyset$; // *fre* is a key-value map to counts the number of activities begin with a certain sensor
3. **for each** sq **in** Sq :
4. $Arr \leftarrow \emptyset$; // *Arr* is a tuple that counts whether a sensor is present
5. $sv \leftarrow getT(sq.s_1) + 1$; // *getT*($sq.s_1$) is employed to get times *t* of $sq.s_1$
6. $update(fre, \{sq.s_1, sv\})$
7. **for each** s **in** sq :
8. $Arr \leftarrow \cup\{(sq.s_1, 1)\}$
9. **end for**
10. $St \leftarrow getDict(SIA, sq.s_1)$; // *getDict*(SIA, sen) is employed to get dict *sen*
11. $St \leftarrow sum(St, Arr)$; // *sum* is add *Arr* to *St*
12. $update(SIA, \{sq.s_1, St\})$
13. **end for**
14. $SIA \leftarrow div(SIA, fre)$; // *div* is used to calculate *SIA* divided by *fre*
15. **return** *SIA*

As shown in Table 1, there are two sensor sequences starting with M004, namely $sq_1 = \{M004, M005, M007, M001, M004, M004, M007, M007\}$, $sq_2 = \{M004, M004, M007, M007, M006, M007, M005, M004\}$, then through Algorithm 1 we can get the importance of the sensor of the activities beginning with M004. The result is $\{(M001, 0.5), (M004, 1), (M005, 1), (M006, 0.5), (M007, 1)\}$, which is represented in *SIA* as $SIA\{(M004, \{(M001, 0.5), (M004, 1), (M005, 1), (M006, 0.5), (M007, 1)\})\}$.

4. Methodology

Our method is mainly divided into two parts: feature extraction and activity recognition. The process is shown in Figure 1.

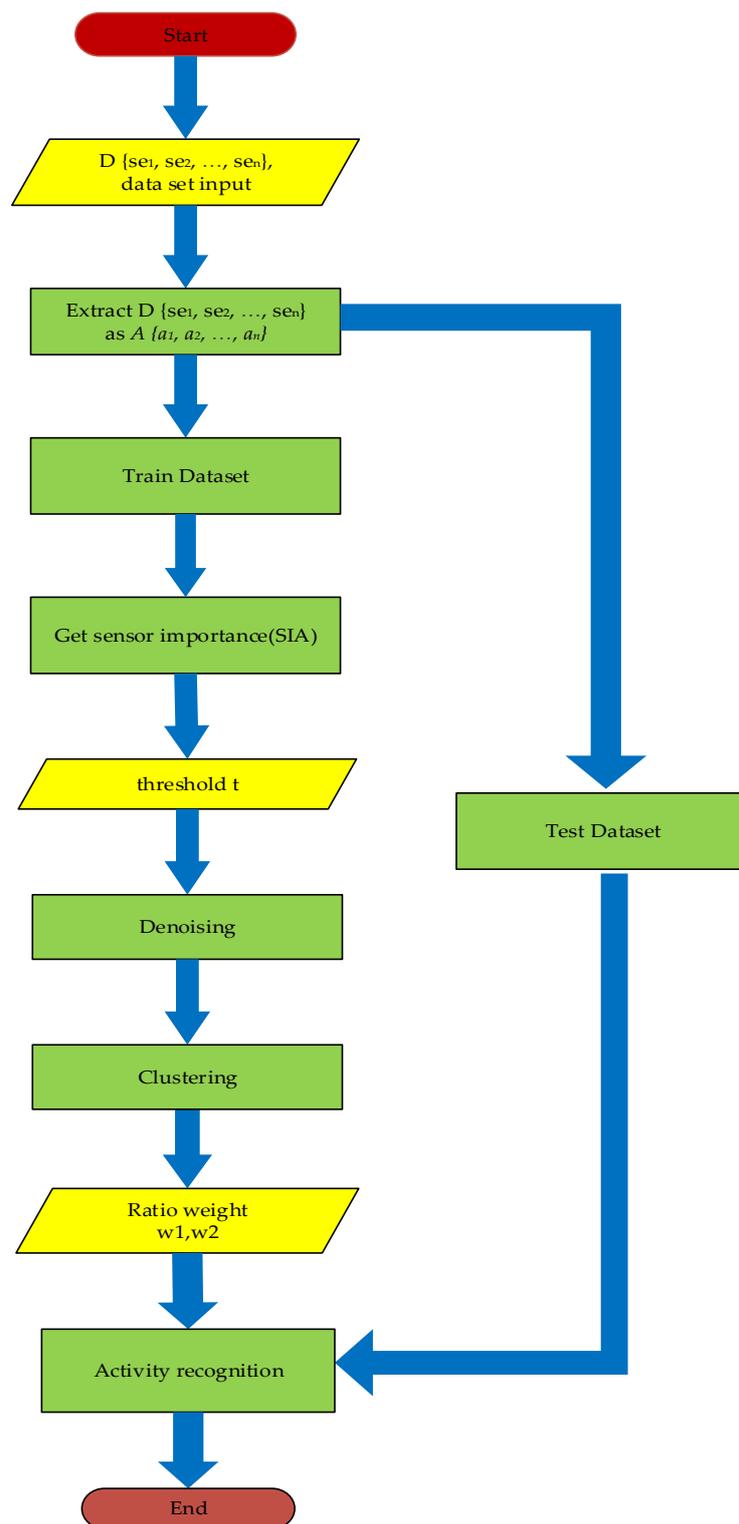


Figure 1. Process for the activity recognition.

4.1. Feature Extraction

The feature extraction method is shown in Algorithm 2. First, we extract the data set $D = \{se_1, se_2 \dots se_n\}$ and extract it as $A = \{a_1, a_2 \dots a_n\}$. Next, we need to de-noise sq in A . It is believed in a two-residential residence, sensor events triggered by one resident may be disturbed by another resident, and these interferences can be removed by our method.

Algorithm 2: Feature Extraction

Input: $A\{a_1, a_2 \dots a_n\}$, a set of activity
 SIA , a tuple of sensor importance
 w , threshold

Output: $Ar\{a_1, a_2 \dots a_n\}$, activities after feature extraction

1. $Ar \leftarrow \emptyset$;
2. **for each** a **in** A :
3. $ns \leftarrow \emptyset$;//new sensor sequence
4. **for each** s **in** $a.sq$:
5. $temp \leftarrow getvalue(SIA, sq.s_1, s)$ //Get the importance of s
6. **if** $temp > w$ **and** s **does not repeat**:
7. $ns \leftarrow \cup s$
8. **end if**
9. **end for**
10. $Ar \leftarrow \cup \{a.bt, a.et, ns\}$
11. **end for**
12. **return** Ar

Our de-noising method has two steps: the first step is to remove duplicate sensor features. It means that if $sq.s_i$ is equal to $sq.s_{i+1}$ in sq , then only $sq.s_i$ is kept. In the second step, we propose a sensor importance measurement method. As shown in Definition 4, SIA counts the importance of other sensors for each sensor that appears at the beginning of the activity. If the importance exceeds a set threshold, we keep the sensor.

Through the above two steps of screening, the extracted feature $Ar = \{bt, et, sq\}$ is finally obtained.

4.2. Activity Recognition

Our activity recognition method is to recognize the activities after extracting features. First, we cluster the activities with the Kmeans based on the bt and et in ar . The number of clusters k was selected using the elbow method. The core index of the elbow method is SSE (sum of the squared errors), the formula is:

$$SSE = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2, \quad (1)$$

where C_i is the i -th cluster, p is the sample point in C_i , m_i is the centroid of C_i (mean of all samples in C_i), and SSE is the clustering error of all samples, which represents the quality of the clustering. The relationship between SSE and k is used to obtain k with the largest curvature in the graph. The specific algorithm is shown in Algorithm 3.

Algorithm 3: SelectK

Input: $Ar\{a_1, a_2 \dots a_n\}$, a set of activity
 num , number of active classes

Output: k , Optimal k value

1. $Val \leftarrow \emptyset$ //
2. $Cal \leftarrow \emptyset$ //
3. $k \leftarrow \emptyset$
4. $Time \leftarrow getTime(Ar)$ //filter out bt and et
5. **for each** n **in** $range(1, num)$:
6. $estimator \leftarrow KMeans(clusters=n)$ //constructing a cluster
7. $estimator.fit(Time)$ //fitting the data
8. $Val \leftarrow getSSE(estimator)$ //get SSE value
9. $Cal \leftarrow \cup \{(n, Val)\}$
10. **end for**
11. $k \leftarrow getCur(Cal)$ //Get k with highest curvature
12. **return** k

After clustering, we first find the cluster that belongs to the input test data, and then calculate the similarity between the instances in the cluster and the test data. Here we propose a similarity matching method. If the test case is $t = \{bt, et, sq\}$, the instance in the training set is $a = \{bt, et, sq\}$, then the similarity between them is expressed as:

$$\text{ratio} = w_1 * \frac{24 - |a.bt - t.bt|}{24} + w_1 * \frac{24 - |a.et - t.et|}{24} + w_2 * \text{Levenshtein.ratio}(a.sq, t.sq), \quad (2)$$

where w_1 and w_2 refer to the weight represented by time and sensor sequence. Here we make $2 * w_1 + w_2 = 1$, 24 refers to 24 h in a day. Levenshtein.ratio is a method to calculate the similarity of sequences, the formula is:

$$\text{Levenshtein.ratio}(sq_1, sq_2) = \frac{\text{sum} - \text{ldist}}{\text{sum}}, \quad (3)$$

where sum refers to the sum of the length of sq_1 and sq_2 , and ldist is the class edit distance.

After obtaining the ratio, select the n closest instances in the cluster and let them vote for the label of test data. The complete recognition algorithm is shown in Algorithm 4.

Algorithm 4: Recognize Activity

Input: $Ar\{a_1, a_2 \dots a_n\}$, a set of activity
 a_t , test sample
 w_1, w_2 , weight of ratio
 n , select the number of training instances with the highest similarity

Output: label, label of a_t

1. $Time \leftarrow \text{getTime}(Ar)$ // get bt and et
2. $AllRatio \leftarrow \emptyset$
3. $k \leftarrow \text{SelectK}(Ar)$ // SelectK is used to select the optimal k
4. $clf \leftarrow \text{KMeans}(n_clusters=k)$ // clustering
5. $clf.fit(Time)$ // fitting the data
6. $m \leftarrow \text{clf.predict}(\{a_t.bt, a_t.et\})$ // find the cluster where the test sample is located
7. **for each** a **in** Ar :
8. **if** $\text{cluster}(a) == m$:
9. $ratio \leftarrow \text{getRatio}(a, a_t, w_1, w_2)$ // calculate ratio
10. $AllRatio \leftarrow \cup ratio$
11. **end if**
12. **end for**
13. $\text{sort}(AllRatio)$ // sort max to min
14. **for** i **in** $\text{range}(n)$:
15. $val \leftarrow \text{getALLRatio}(i)$ // get the first n instances of maximum ratio
16. $topk_y \leftarrow \cup val$
17. **end for**
18. $label \leftarrow \text{vote}(topk_y)$ // vote for label
19. **return** label

5. Results and Evaluation

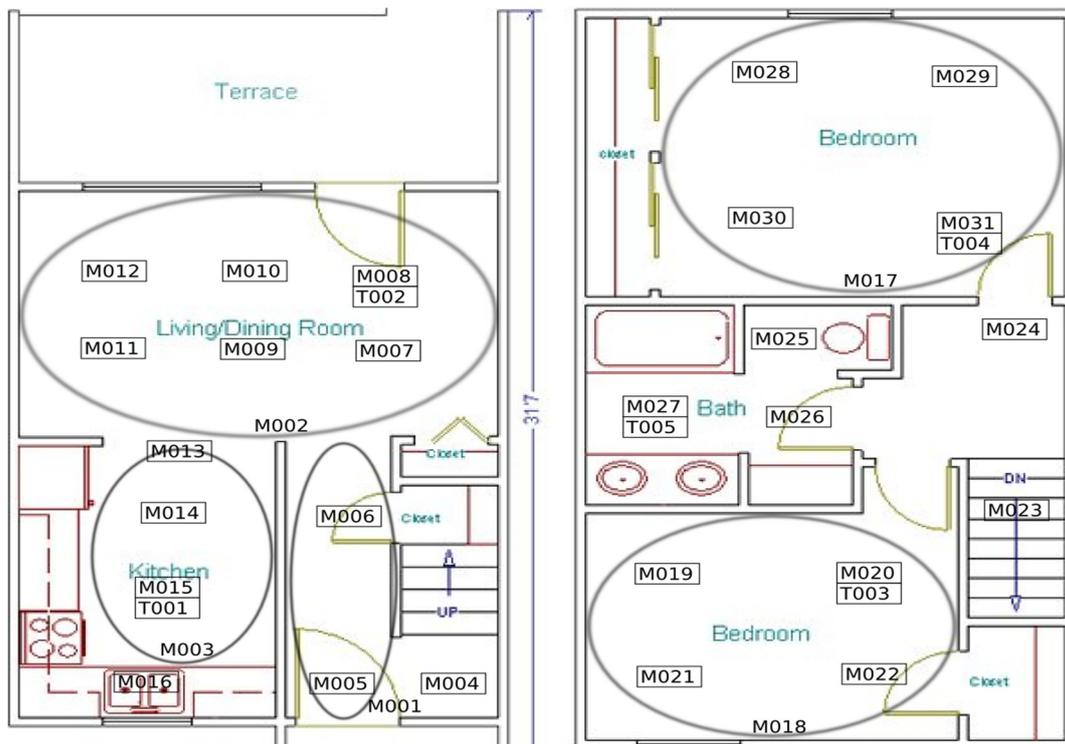
5.1. Datasets

We used two two-resident datasets from Washington State University for experiments [38]; they are “Tulum2010” and “Cairo”. Details of these two datasets are listed in Table 2.

Table 2. Statistical information concerning datasets “Tulum2010” and “Cairo.”.

	Sensors	Activity Categories	Activity Instances	Residents	Measurement Time
Tulum2010	36 (2 categories)	14	7980	2	98 days
Cairo	32 (2 categories)	13	600	2	57 days

The Tulum2010 dataset was collected in the Washington State University (WSU) Tulum smart apartment during 2009–2010. The apartment housed two married residents who performed normal daily activities. The sensor layout is shown in Figure 2. We selected the first three months of the data set.

**Figure 2.** Tulum2010 sensor layout.

Tulum2010 dataset contains 31 motion sensors (M001 through M031) and 5 temperature sensors (T001 to T005). Its 14 activities are “Bathing”(B), “Eating”(E), “Bed_Toilet_Transition”(“B_T”), “Watch_TV”(“W_TV”), “Work_Table”(“W_T”), “Work_LivingRm”(“W_L”), “Enter_Home”(“E_H”), “Leave_Home”(“L_H”), “Personal_Hygiene”(“P_H”), “Sleeping_in_Bed”(“S_B”), “Yoga”(“Y”), “Work_Bedroom_1”(“W_B1”), “Work_Bedroom_2”(“W_B2”), “Meal_Preparation”(“M_P”).

The Cairo dataset was collected in the home of a voluntary adult couple. A couple and a dog live in the smart apartment. The couple’s children also visited the house at least once. The sensor layout is shown in Figure 3.

Cairo dataset contains 27 motion sensors (M001 to M027) and 5 temperature sensors (T001 to T005). Its 13 activities are “Bed_to_toilet”(“B_T”), “Breakfast”(“B”), “Night_wandering”(“N_W”), “R2_wake”(“R2_W”), “Leave_home”(“L_H”), “R2_take_medicine”(“T_M”), “R1_sleep”(“R1_S”), “Lunch”(“LUN”), “Laundry”(“L”), “R1_wake”(“R1_W”), “R2_sleep”(“R2_S”), “Dinner”(“D”), “R1_work_in_office”(“W_O”).

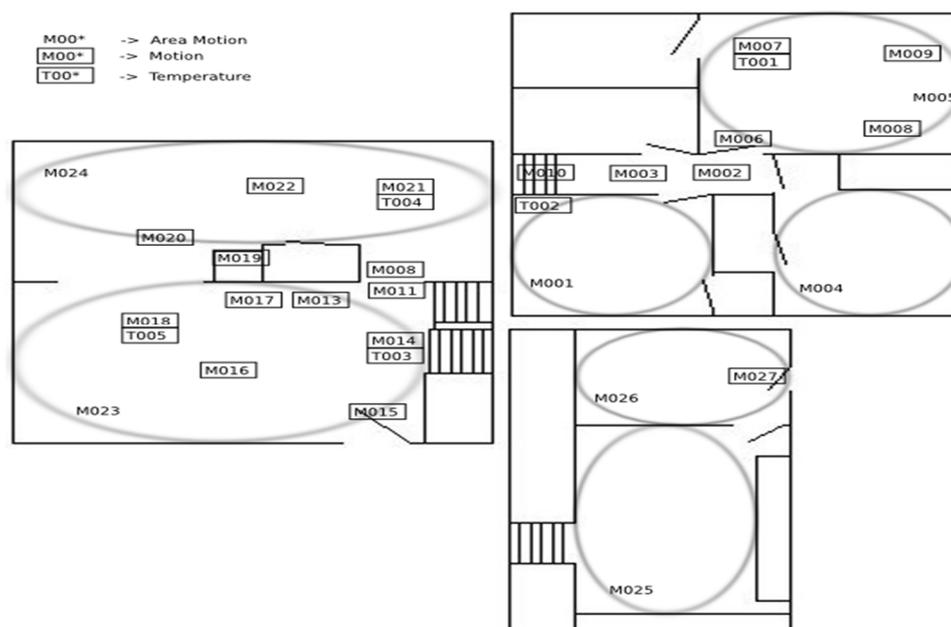


Figure 3. Cairo sensor layout.

5.2. Evaluation Method

In this section, we mainly applied all the examples in the considered dataset and performed five cross-validations to evaluate the performance of our method. During the process, it was performed five times. In each iteration, four folds are selected for training and one fold is used for testing. The final accuracy is the average of the five results. Except for the accuracy, in the paper we also used others to evaluate the results, such as the precision, the recall, and the F-Measures. We used the confusion matrix in Table 3 to represent the number of true positives, true negatives, false positives, and false negatives for the 2 classification problems.

Table 3. Confusion matrix presenting number of true positives, true negatives, false positives, and false negatives for a 2-class classification problem.

		Actual Class	
		1	2
Predicted Class	1	TP(true positive)	FP(false positive)
	2	FN(false negative)	TN(true negative)

The columns and rows in the matrix refer to the actual and predicted classes by a classification model, respectively. Based on the confusion matrix the precision, recall and F-Measure for class 1 can be calculated as presented in Equations (4)–(7), respectively:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F-Measure} = \frac{2 * TP}{2 * TP + FP + FN} \quad (7)$$

The four measures were calculated for each class, taking into account the class imbalance problem; the final result is presented as the average of all classes.

5.3. Results and Evaluation

We first compare our results with several common single classifiers, such as K-Nearest Neighbor (KNN), lib Support Vector Machine (libSVM), Sequential Minimal Optimization (SMO), Naïve Bayes (NB), PIPPER, C4.5 and Random Forests (RF). In addition to these, we also compare several more complex classifiers. These classifiers are all implemented in Weka, and the specific results are shown in Tables 4 and 5.

Table 4. Tulum2010 performance.

	Accuracy	Precision	Recall	F-Measure
KNN	77.10%	77.20%	77.10%	77.10%
LibSVM	72.40%	75.90%	72.40%	71.70%
SMO	54.20%	61.60%	54.20%	50.10%
NB	50.60%	65.50%	50.60%	52.60%
RIPPER	80.30%	80.50%	80.30%	80.30%
C4.5	80.90%	81.00%	80.90%	80.90%
RF	83.60%	83.50%	83.60%	83.40%
Our Method	88.10%	88.00%	88.10%	87.90%

Table 5. Cairo performance.

	Accuracy	Precision	Recall	F-Measure
KNN	80.20%	81.40%	80.20%	80.40%
LibSVM	30.30%	38.50%	30.30%	27.00%
SMO	71.70%	76.90%	71.70%	68.40%
NB	79.20%	81.70%	79.20%	79.50%
RIPPER	76.80%	78.30%	76.80%	76.90%
C4.5	83.00%	83.20%	83.00%	83.00%
RF	89.60%	89.90%	89.70%	89.70%
Our Method	92.00%	92.90%	92.00%	92.00%

It can be seen from Tables 4 and 5 that our method and RF are the highest performance methods, and our method is better than RF in terms of performance. To better understand the performance of our method, we construct a confusion matrix. Tables 6 and 7 show the confusion matrices for Tulum2010 and Cairo.

From Table 6, we can see that in the Tulum2010 dataset we have almost no errors in the recognition of B, B_T, M_P, P_H, S_B, W_B1, W_B2, and other activities. However, the recognition effect for E_H and L_H, E and W_T, W_TV, and W_L is a bit poor, which may be because these activities are represented by some of the same sensors compared to other activities, making them more difficult to recognize. For example, E_H and L_H both occur at the door, W_T and E both occur in the dining room, and W_TV and W_L both occur in the bedroom.

Compared with Tulum2010, the Cairo dataset has fewer activities. From Table 7, we can see that there are three misidentifications of N_W activity as B_T. This may be because the routes and the occurrence times of these three N_W are similar to B_T. However, it can be seen that the overall recognition effect of our method is still very good.

Table 6. Confusion matrix for Tulum2010.

	B	B_T	E	E_H	L_H	M_P	P_H	S_B	W_TV	W_B1	W_B2	W_L	W_T	Y
B	80						1							
B_T		16					6				1			
E			62		1	3							27	
E_H				8	5				1		1			
L_H				4	9									2
M_P			5			192							1	
P_H	1	2					169							
S_B								36			2			
W_TV									278		1	45		
W_B1										94				
W_B2										1	267			
W_L									54			60		
W_T			19			2							136	
Y									3					2

Table 7. Confusion matrix for Cairo.

	B_T	B	D	L	L_H	L	N_W	R1_S	R1_W	W_O	R2_S	T_M	R2_W
B_T	6												
B		8			1								1
D			9										
L				1	1								
L_H					13					1			
LUN						8							
N_W							11						
R1_S		3					1	9					
R1_W		1							10				
W_O										10			
R2_S								1			10		
T_M												9	
R2_W													11

5.4. Parameter Selection

5.4.1. Selection of k

Our k -value selection method is to use the elbow method. Its formula is shown in Equation (1). The core idea of the elbow method is: as the number of clusters k increases, the sample division will become more refined, and the degree of aggregation of each cluster will gradually increase, so the SSE will naturally become smaller. Moreover, when k is less than the number of true clusters, the increase of k will greatly make the degree of aggregation of each cluster increasing, so the decline of SSE will be quick. When k reaches the true number of clusters, the increase of the aggregation degree obtained by increasing k will be slow, so the decline of SSE will decrease sharply, and then gradually flatten as the value of k continues to increase. In other words, the relationship between SSE and k is the shape of an elbow, and the value of k , this elbow is the true cluster number of the data. Of course, this is why the method is called the elbow method.

Figure 4 shows the relationship between k and SSE in the Tulum2010 dataset. It can be seen that the curvature is the largest when $k = 3$, so we choose $k = 3$ as the optimal number of clusters.

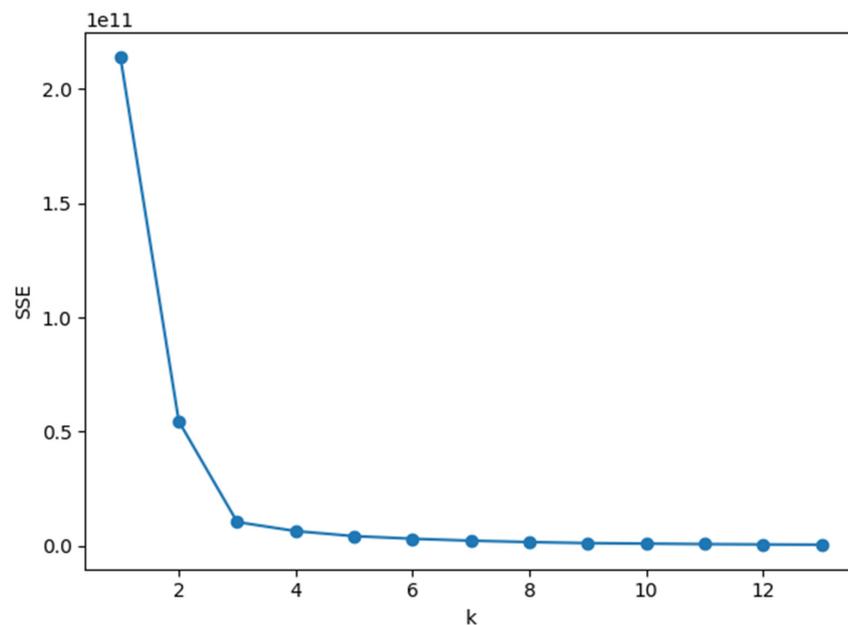


Figure 4. Relationship between k and sum of the squared errors (SSE) in Tulum2010.

5.4.2. Selection of n and w_1, w_2 Values

We believe that the choice of n value should be less than the number of kinds of activities in the data set, which makes our results more accurate. Equation (2) gives a detailed similarity formula. For w_1 and w_2 , we always keep $2 * w_1 \leq w_2$ and $2 * w_1 + w_2 = 1$. Figure 5 shows the relationship between n and accuracy when $w_1 = 0.15$ and $w_2 = 0.7$ in the Tulum2010 data set. As you can see, the fluctuation of accuracy is small.

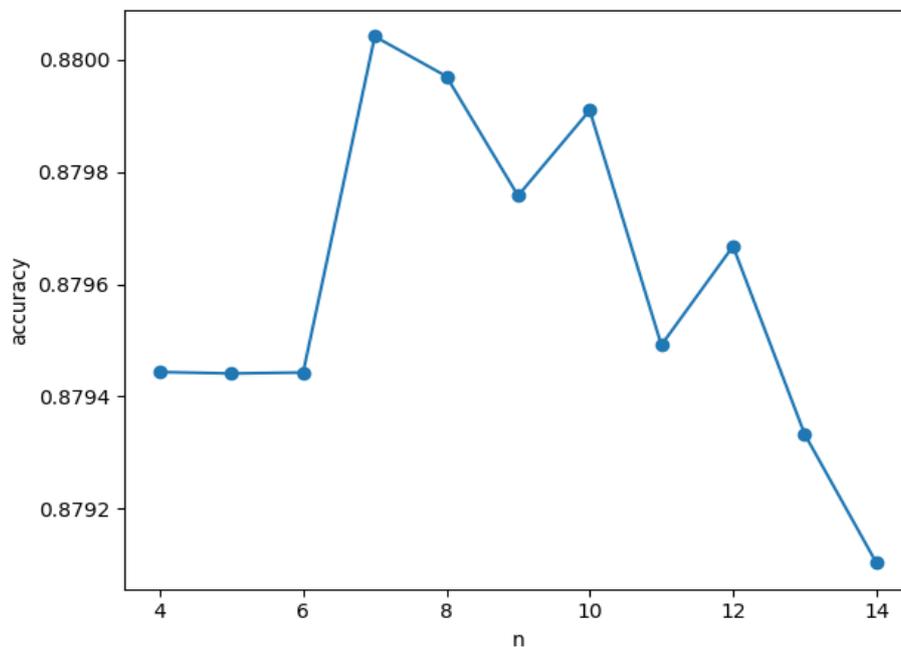


Figure 5. Relationship between n and accuracy when weight of ratio (w_1) = 0.15 and w_2 = 0.7.

6. Conclusions

This paper presents a daily activity recognition method based on time clustering for two residents in a smart home. First, noise reduction processing is performed on the features. Second, cluster is

performed to separate activities that occur at the same space but at different times. Finally, a similarity matching formula based on Levenshtein Distance is proposed for daily activity recognition. The proposed method not only reduces the interference caused by the activities of different residents in daily life, but also separates the activities of residents at different times in the same space. We evaluated the proposed on two public datasets, Tulum2010. The results show that our method works well on large and small datasets.

Author Contributions: Conceptualization, J.G. and Y.L.; methodology, J.G. and Y.L.; validation, M.H. and J.R.; and S.H.; writing-review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61976124); the Fundamental Research Funds for the Central Universities (No. 3132019201); the Open Project Program of Artificial Intelligence Key Laboratory of Sichuan Province (No. 2018RYJ09).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krishnan, N.C.; Cook, D. Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* **2014**, *10*, 138–154. [[CrossRef](#)]
2. Xia, J.; Siochi, A. A real-time respiratory motion monitoring system using KINECT: Proof of concept. *Med. Phys.* **2012**, *39*, 2682–2685. [[CrossRef](#)]
3. Qin, S.; Zhu, X.; Yang, Y. Real-time hand gesture recognition from depth images using convex shape decomposition method. *J. Signal Process Syst.* **2014**, *74*, 47–58. [[CrossRef](#)]
4. Dai, G.; Hu, Y.; Yang, Y.; Zhang, N.; Abraham, A.; Liu, H. A Novel Fuzzy Rule Extraction Approach Using Gaussian Kernel Based Granular Computing. *Knowl. Inf. Syst.* **2019**, *61*, 821–846. [[CrossRef](#)]
5. Ni, Q.; García Hernando, A.; La, L. The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development. *Sensors* **2015**, *15*, 11312–11362. [[CrossRef](#)]
6. Chen, Z.; Zhu, Q.; Chai, S.Y.; Zhang, L. Robust human activity recognition using smartphone sensors via CT-PCA and online SVM. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3070–3080. [[CrossRef](#)]
7. Yang, L.; Yang, Y.; Mgaya, G.B.; Zhang, B.; Chen, L.; Liu, H. Novel Fast Networking Approaches Mining Underlying Structures from Investment Big Data. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**. [[CrossRef](#)]
8. Gupta, P.; Dallas, T. Feature selection and activity recognition system using a single triaxial accelerometer. *IEEE Trans. Bio-Med. Eng.* **2014**, *61*, 1780–1786. [[CrossRef](#)]
9. Liu, Y.; Wang, X.X.; Zhai, Z.G.; Chen, R.; Zhang, B.; Jiang, Y. Timely daily activity recognition from headmost sensor events. *ISA Trans.* **2019**, *94*, 379–390. [[CrossRef](#)]
10. Kim, S.S.; McLoone, S.; Byeon, J.H.; Lee, S.; Liu, H. Cognitively-inspired artificial bee colony clustering for cognitive wireless sensor networks. *Cogn. Comput.* **2017**, *9*, 207–224. [[CrossRef](#)]
11. Liu, Y.Q.; Yi, X.K.; Chen, R.; Zhai, Z.G.; Gu, J.G. Feature Extraction Based on Information Gain and Sequential Pattern for English Question Classification. *IET Soft.* **2018**, *12*, 520–526. [[CrossRef](#)]
12. Chen, L.; Nugent, C.D.; Wang, H. A knowledge-driven approach to activity recognition in smart homes. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 961–974. [[CrossRef](#)]
13. Ye, J.; Stevenson, G. Semantics-driven multi-user concurrent activity recognition. In Proceedings of the International Joint Conference on Ambient Intelligence, Dublin, Ireland, 3–5 December 2013; pp. 204–219.
14. Ye, J.; Stevenson, G.; Dobson, S. Kcar: A knowledge-driven approach for concurrent activity recognition. *Pervasive Mob. Comput.* **2015**, *19*, 47–70. [[CrossRef](#)]
15. Wemlinger, Z.E.; Holder, L.B. Cross-environment activity recognition using a shared semantic vocabulary. *Pervasive Mob. Comput.* **2018**, *51*, 150–159. [[CrossRef](#)]
16. Liu, Y.Q.; Mu, Y.; Chen, K.Y.; Li, Y.M.; Guo, J.H. Daily Activity Feature Selection in Smart Homes Based on Pearson Correlation Coefficient. *Neural Process. Lett.* **2020**. [[CrossRef](#)]
17. Nishkam, R.; Nikhil, D.; Preetham, M.; Michael, L.L. Activity recognition from accelerometer data. In Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence, Pittsburgh, PA, USA, 9–13 July 2005; pp. 1541–1546.

18. Rubén, S.; Juan, M.M.; José, M.P. HMM Adaptation for Improving a Human Activity Recognition System. *Algorithms* **2016**, *9*, 60.
19. Asghari, P.; Nazerfard, E. Activity Recognition Using Hierarchical Hidden Markov Models on Streaming Sensor Data. In Proceedings of the 9th International Symposium on Telecommunications, Tehran, Iran, 17–19 December 2018; pp. 416–420.
20. Hu, C.; Chen, Y.; Hu, L.; Peng, X. A novel random forests based class incremental learning method for activity recognition. *Pattern Recognit.* **2018**, *78*, 277–290. [[CrossRef](#)]
21. Jurek, A.; Nugent, C.; Bi, Y.; Wu, S. Clustering-Based Ensemble Learning for Activity Recognition in Smart Homes. *Sensors* **2014**, *14*, 12285–12304. [[CrossRef](#)]
22. Jurek, A.; Bi, Y.; Wu, S.; Nugent, C.D. A Cluster-Based Classifier Ensemble as an Alternative to the Nearest Neighbour Ensemble. In Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence, Athens, Greece, 7–9 November 2012; pp. 1100–1105.
23. Arifoglu, D.; Bouchachia, A. Detection of abnormal behaviour for dementia sufferers using Convolutional Neural Networks. *Artif. Intell. Med.* **2019**, *94*, 88–95. [[CrossRef](#)]
24. Gochoo, M.; Tan, T.; Liu, S.; Jean, F.; Alnajjar, F.S.; Huang, S. Unobtrusive Activity Recognition of Elderly People Living Alone Using Anonymous Binary Sensors and DCNN. *IEEE J. Biomed. Health Inform.* **2019**, *23*, 693–702. [[CrossRef](#)]
25. Medina-Quero, J.; Zhang, S.; Nugent, C.; Espinilla, M. Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition. *Expert Syst. Appl.* **2018**, *114*, 441–453. [[CrossRef](#)]
26. Hu, D.H.; Zheng, V.W.; Yang, Q. Cross-domain activity recognition via transfer learning. *Pervasive Mob. Comput.* **2011**, *7*, 344–358. [[CrossRef](#)]
27. Wen, J.; Zhong, M. Activity discovering and modelling with labelled and unlabelled data in smart environments. *Expert Syst. Appl.* **2015**, *42*, 5800–5810. [[CrossRef](#)]
28. Gu, T.; Wu, Z.; Tao, X.; Pung, H.K.; Lu, J. epSICAR: An emerging patterns based approach to sequential interleaved and concurrent activity recognition. In Proceedings of the 7th IEEE International Conference on Pervasive Computing and Communications, Galveston, TX, USA, 9–13 March 2009; pp. 1–9.
29. Rashidi, P.; Cook, D.J. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **2009**, *39*, 949–959. [[CrossRef](#)]
30. Huynh, T.; Schiele, B. Unsupervised discovery of structure in activity data using multiple eigenspaces. In Proceedings of the 2nd International Workshop on Location- and Context-Awareness, Dublin, Ireland, 10–11 May 2006; pp. 151–167.
31. Barger, T.S.; Brown, D.E.; Alwan, M. Health-status monitoring through analysis of behavioral patterns. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **2005**, *35*, 22–27. [[CrossRef](#)]
32. Dimitrov, T.; Pauli, J.; Naroska, E. Unsupervised recognition of ADLs. In Proceedings of the 6th Hellenic Conference on AI, Athens, Greece, 4–7 May 2010; pp. 71–80.
33. Palmes, P.; Pung, H.K.; Gu, T.; Xue, W.; Chen, S. Object relevance weight pattern mining for activity recognition and segmentation. *Pervasive Mob. Comput.* **2010**, *6*, 43–57. [[CrossRef](#)]
34. Hao, J.; Bouzouane, A.; Gaboury, S. Recognizing multi-resident activities in non-intrusive sensor-based smart homes by formal concept analysis. *Neurocomputing* **2018**, *318*, 75–89. [[CrossRef](#)]
35. Alemdar, H.; Ersoy, C. Multi-resident activity tracking and recognition in smart environments. *J. Ambient Intell. Humanized Comput.* **2017**, *8*, 513–529. [[CrossRef](#)]
36. Guo, J.; Mu, Y.; Xiong, M.; Liu, Y.; Gu, J. Activity Feature Solving Based on TF-IDF for Activity Recognition in Smart Homes. *Complexity* **2019**, *2019*. [[CrossRef](#)]
37. Lu, L.; Cai, Q.; Zhan, Y. Activity Recognition in Smart Homes. *Multimed. Tools Appl.* **2016**, *76*, 24203–24220. [[CrossRef](#)]
38. WSU CASAS Datasets. 2016. Available online: <http://ailab.wsu.edu/casas/datasets.html> (accessed on 2 February 2016).

