energies

MDPI

*Article*

# Activity-Aware Energy-Efficient Automation of Smart Buildings

**Brian L. Thomas and Diane J. Cook** *

School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99163, USA; bthomas@eecs.wsu.edu

* Correspondence: cook@eecs.wsu.edu; Tel.: +1-509-335-4985

Academic Editor: Hossam A. Gabbar (Gaber)

Received: 3 June 2016; Accepted: 4 June 2016; Published: 9 August 2016

**Abstract:** This paper introduces the idea of activity-aware cyber-physical systems (CPS). Activity-aware systems allow smart city services to adapt to the needs of individual residents by being sensitive to their daily tasks. The paper first defines activity recognition and activity prediction algorithms that form the foundation of activity-aware CPS and implement a prototype activity-aware building automation system, called CASAS activity aware resource learning (CARL). Evaluation of CARL on real sensor data shows not only an accurate ability to sense and predict activities but an effective means of automation buildings that reduces energy consumption while being sensitive to user activities in the building. Our ideas are demonstrated in the context of a smart home but can be utilized in a variety of smart city settings including smart offices, smart hospitals, and smart communities.

**Keywords:** smart cities; cyber-physical systems (CPS); activity recognition; activity prediction; building automation

## 1. Introduction

In recent years, cyber-physical systems (CPS) have been enhanced by the notion of context-aware computing. Sensing the current situation and reasoning about its implication can improve the design of the physical system and enhance its real-time system resiliency and responsiveness. In this paper, CPS are pushed to this level by introducing the notion of activity-aware CPS. Deploying activity-aware CPS requires several computational components to make them aware of user activities. Such CPS need a method of identifying current activities (activity recognition), as well as a method of forecasting when activities are going to begin and end (activity prediction).

Activity-aware systems are valuable when transforming cities to smart cities because the services, such as building automation, transportation routing, and energy provisioning, can now adapt to the needs of individual users. The hypothesis of this paper is that smart buildings, a key component of smart cities, can benefit from being activity aware. This hypothesis is validated in the context of a CPS that automates building control for energy efficiency. Our activity-aware smart automation system, CARL (CASAS activity aware resource learning), is built on the foundation of the CASAS smart environment infrastructure [1]. Data are collected from sensors embedded in everyday building settings found in smart cities, such as smart offices, smart hospitals, and smart homes. The collected data are used to identify activities that residents are performing and to determine the devices that are used in the context of those activities.

The goal of CARL is to automate a smart building by turning off devices that are not needed for the current activity and leaving on devices that are required. By recognizing the current activity, a building found in a smart city is sensitive to its residents and does not turn off devices that they need. User adaptation is then further enhanced by predicting when the current activity will end and the next begin. By providing this activity-aware energy-efficient building automation, smart cities can realize energy savings while still meeting the needs of the individuals who live and work there. To validate our notion of an activity-aware energy-efficient building, CARL is evaluated to determine its ability to efficiently automate an actual smart building without disrupting resident activities.

## 2. Energy-Efficient Smart Buildings

The impact of lifestyle choices on energy usage and the environment are becoming increasingly noticeable and therefore a focus of resource on building automation and smart cities. As a result, research attention is being directed toward green technology, environmentally-friendly building design, and active demand response within the smart grid. This article examines the behavior side of sustainability and introduces ubiquitous computing technologies that may aid in reducing energy consumption. In particular, an activity-aware intervention is described that promotes energy efficient, sustainable building automation.

In 2015, the United States consumed 97.651 quadrillion BTU of energy, a 300% increase from 1949 [2]. The growth of energy usage is not entirely due to manufacturing plants and automobiles: residential and commercial buildings are responsible for 40% of the energy consumption [3]. There exists evidence that residential consumer behavior can be influenced to be more sustainable. For example, home residents have reduced consumption by as much as 15% in response to simply viewing raw usage data [4]. Changing behavioral patterns in these environments can influence usage by as much as 90% in commercial buildings and 100% in household settings [5].

Until recently, occupant behavior has been difficult to accurately capture. Self-reports of behavior and energy consumption is error prone for some populations [6] and whole-home meter monitoring does not capture the behaviors in the home that influence consumption. Approaches have been utilized to explore the gap between the minimum amount of consumption that is needed for daily activities and the consumption that is actually observed [7]. Some early work has focused on linking resident activity with energy consumption. The hypothesis that providing users with knowledge about the relationship between their activities and energy consumption and automation support for energy reduction will result in substantial decreases in overall consumption is supported by an increasing body of work that links awareness of energy consumption and its impact on behavioral routines and behavioral change [8–10]. Until recently, validating this hypothesis was not possible. However, with the convergence of technologies in ubiquitous computing and machine learning, gathering data on human behavior is now automatable. Data can be collected from sensor-filled smart buildings and smart phones in an unobtrusive manner while individuals perform their normal daily routines. Because these sensor modalities operate in a continuous mode, feedback and interventions repeat ad infinitum, thereby maximizing the persistence effect.

In this paper, information from sensor-filled environments is utilized to intelligently automate a smart building. Automating control of buildings for energy efficiency has been explored by other groups [11]. However, this work represents the first known approach in which activity awareness is used to more intelligently automate the environment.

## 3. Smart Environments

Computers are commonly embedded in familiar objects such as home appliances and mobile devices, gradually pervading almost every level of society. In the last decade, machine learning and pervasive computing technologies have matured to the point where this power is not only integrated with our lives but it can provide context aware, automated support in our everyday environments. One physical embodiment of such a system is a smart home. In the home or other smart building environment, computer software that plays the role of an intelligent agent perceives the state of the physical environment and residents using sensors, reasons about the state of the environment using artificial intelligence techniques, and then takes actions to achieve specified goals.

Activity-aware building automation CPS can be accomplished with any sensor-filled physical environment (Figure 1). One physical embodiment of such a system is a smart home. In the home environment, computer software that plays the role of an intelligent agent perceives the state of the physical environment and residents using sensors, reasons about this state using machine learning and data mining, and then takes actions to achieve specified goals.

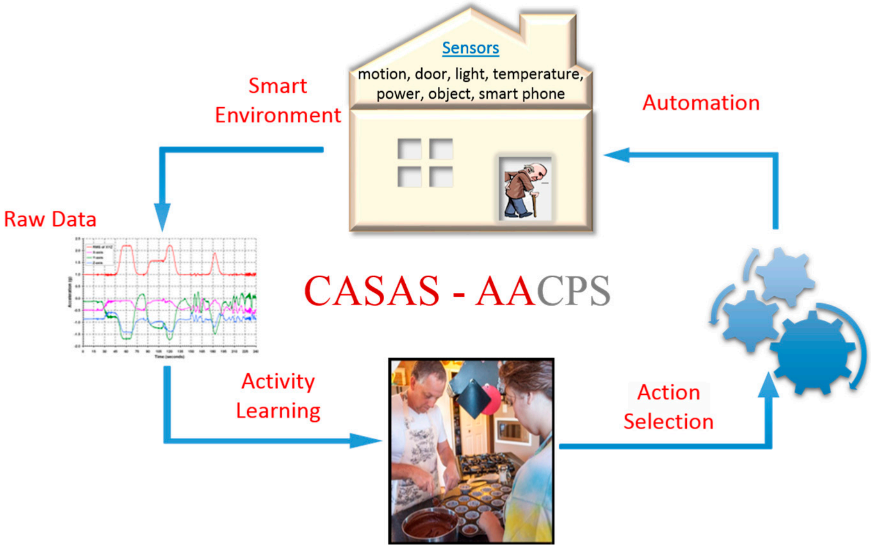


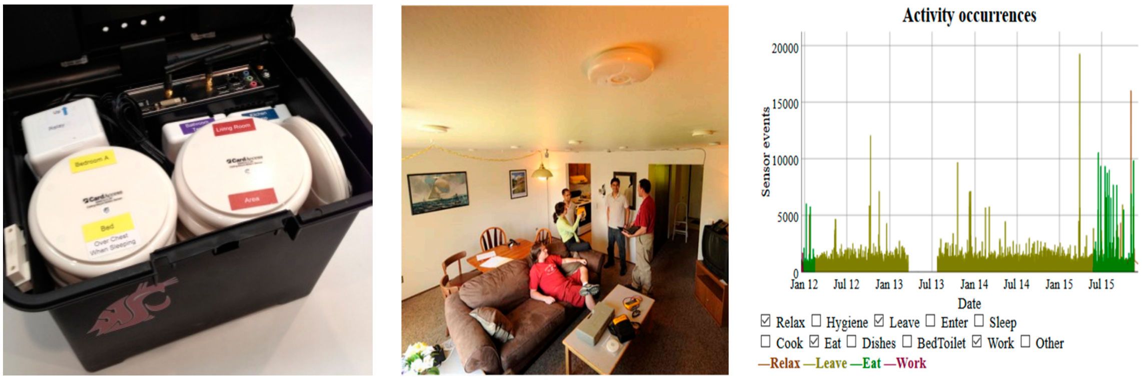
**Figure 1.** The physical system (sensors, home) work together with humans and computational components (activity learning) to provide activity-aware automation.

Smart home technology is being increasingly recognized valuable for applications including health monitoring and home automation [12]. Smart home projects, including the Aware home [13], the Gator Tech smart home [14], and the MavHome [15], demonstrated the capabilities of using sensors and computers to create a home that reasons about its state and takes actions to make the home more comfortable. Smart homes have recently been a focus for companies including GE, Intel, iControl, Control4, Brillo, and Google, who are creating smart home operating systems, interfaces, developer platforms, and maintenance plans for the consumer. Many of these projects provide a basic infrastructure for collecting sensor data and automating devices. The key to making such environments intelligent is the software that reasons about the home using techniques, such as activity recognition and activity prediction, as is the focus in this paper.

In this paper, the system is implemented and evaluated in the context of a CASAS smart home. Due to the difficulty of creating a fully-functional smart environment infrastructure, many of the early smart home projects described in the previous paragraph are tested on simulated or lab-based data [16,17]. To support the scaling of smart environment research, a streamlined "smart home in a box" (SHiB) was designed [1], shown in Figure 2. SHiB components communicate via bridges. Bridges are created for Zigbee communication, for archiving sensor messages in a relational database, and for each application. While each site runs independently, the smart building site also securely upload events to be stored in a relational database in the cloud.

Data has been collected in 80 smart environment sites to date. This paper highlights one such site, a smart home environment named Navan. Navan is a single-resident apartment with a floor plan, shown in Figure 3. Navan is equipped with 118 sensors. To track the location of smart home residents we place infrared motion sensors on the ceilings with removable adhesive strips. Most of the motion sensors are focused to sense area in a one-meter diameter area immediately below the sensor. However, additional motion sensors are placed in each major room, which have a much broader coverage in order to indicate whether human (or pet) motion is occurring anywhere in the room. The circles in

Figure 3 represent the positions of the motion sensors. The square icons in the figure indicate the presents of magnetic door sensors, which register the open/shut status of external doors as well as cabinets in the kitchen and bathrooms. Coupled with these are additional sensors that monitor ambient light and ambient temperature, which are useful for recognizing key activities such as bathing and cooking and for sensing internal (and to an extent, external) weather conditions. Additionally, Navan also includes temperature-only sensors (represented as stars in the figure) that are placed in pairs throughout the apartment at 8″ from the ceiling and 12″ from the floor to identify temperature gradients. Electricity usage data are collected in Navan using a Ted5000 power meter that provides instantaneous usage wattages every few seconds. Arduino-based WiFi thermostats (represented by hexagonal icons in the figure) were designed, built, and installed to monitor use of the baseboard heaters in individual rooms and to log temperature setpoints.



(**a**) (**b**) (**c**)

**Figure 2.** (**a**) Smart home in a box; (**b**) smart apartment; and (**c**) activity graph.

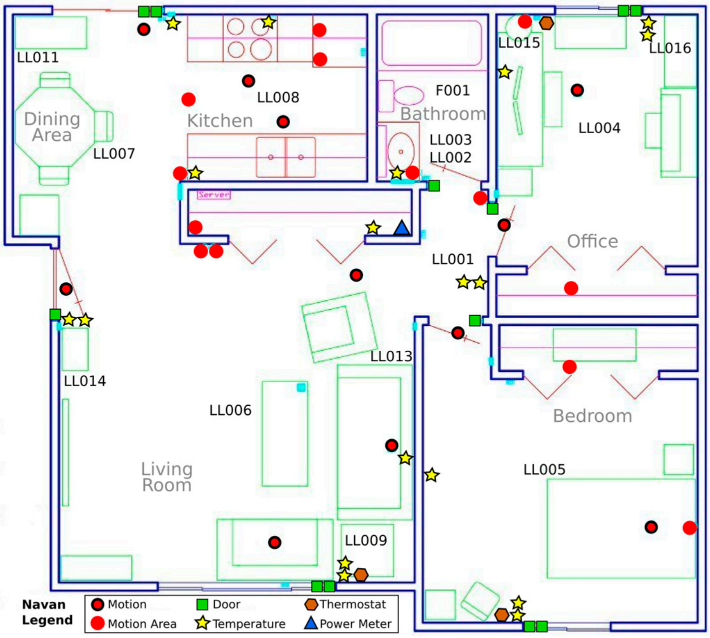
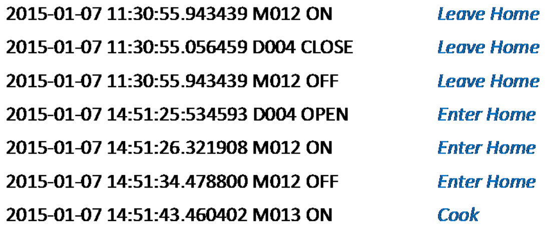


**Figure 3.** Navan automated smart home testbed.

The sensors in the smart home are discrete event sensors. When a state change is sensed (e.g., there is motion in the area, a cessation of motion in the area, a significant temperature change, or a change in door status), the sensor generates a reading that is sent (as a text message) to the smart home middleware. The middleware logs the ID of the sensor generating the reading together with the date and time of the reading and the state of the sensor. Figure 4 shows a sample of the readings that are generated by one such smart home.



```
2015-01-07 11:30:55.943439 M012 ON          Leave Home
2015-01-07 11:30:55.056459 D004 CLOSE       Leave Home
2015-01-07 11:30:55.943439 M012 OFF         Leave Home
2015-01-07 14:51:25:534593 D004 OPEN        Enter Home
2015-01-07 14:51:26.321908 M012 ON          Enter Home
2015-01-07 14:51:34.478800 M012 OFF         Enter Home
2015-01-07 14:51:43.460402 M013 ON          Cook
```

**Figure 4.** Example text-based sensor data. Sensor IDs starting with M are motion sensors and IDs starting with D are door sensors. Sensor M012 is located near the external door, as is sensor D004. Sensor M013 is located in the kitchen.

To facilitate control of devices inside Navan, ZigBee light switches are installed to control lights and to the bathroom fan. In addition, custom electrical boxes are designed with ZigBee light switches, as shown in Figure 5, to monitor and control additional devices including reading lamps and speakers. Each light switch reports changes in state of the device as well as button taps and tap counts. These taps provide a mechanism for the resident to provide feedback to the home automation system. In Figure 3, the locations of devices that are controlled by the ZigBee light switches are indicated by the name of each device. All of the indicated devices represent lights or lamps except for F001 (the bathroom fan) and LL014 (the television speakers).

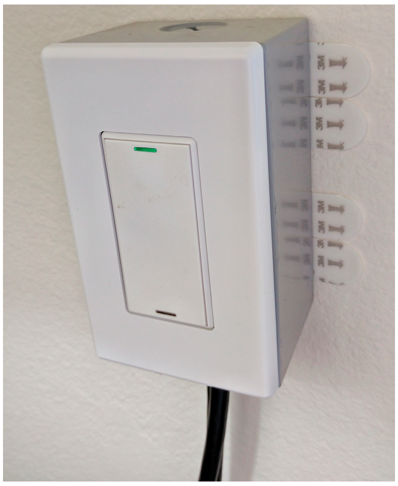


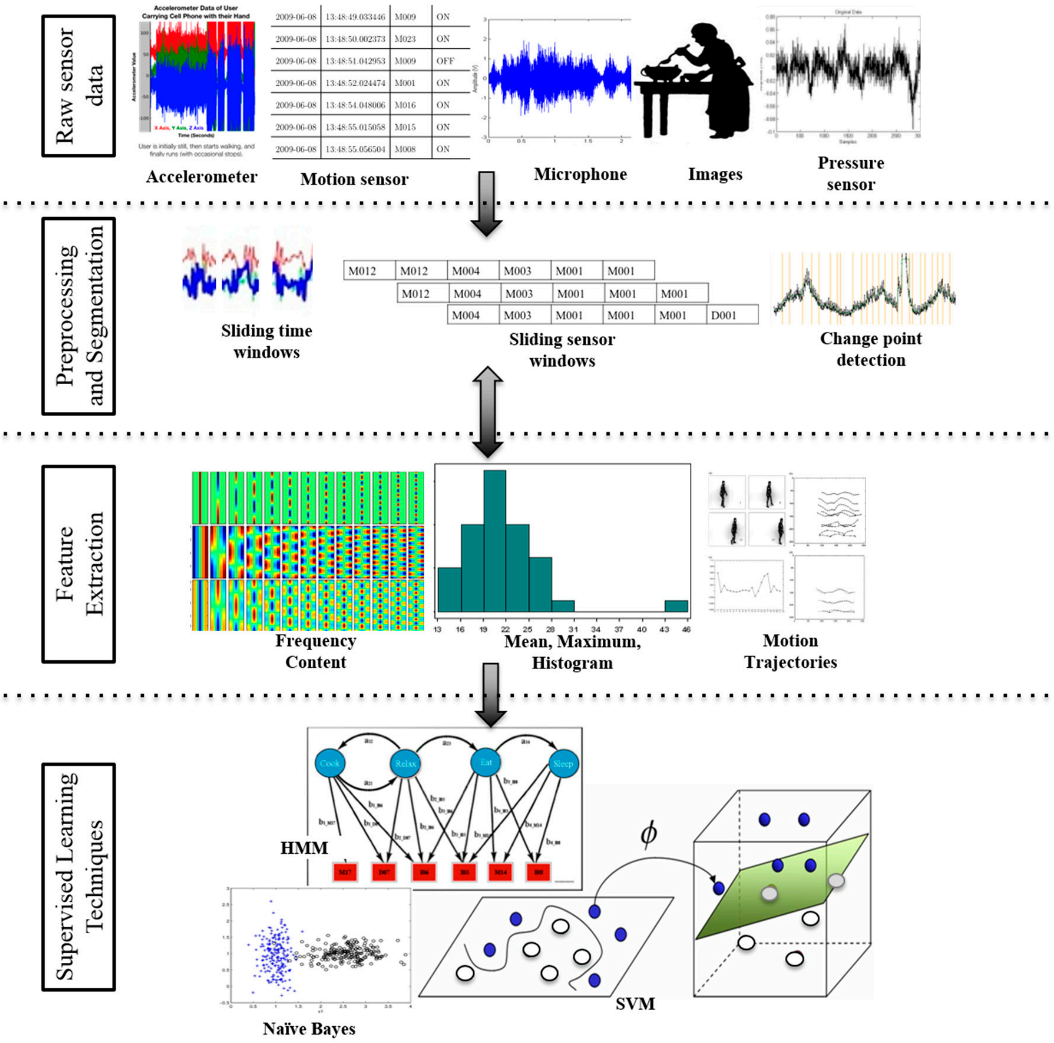
**Figure 5.** A ZigBee light switch is used to control devices and provide user feedback.

## 4. Activity Awareness

Learning and understanding observed activities is at the center of many fields of study and is essential for CPS such as smart buildings that are sensitive to the needs of the humans they serve. An individual’s activities affect that individual, those around him/her, society, and the environment. CPS that operate in real-world complex applications such as building automation require the depth of information that is provided by activity learning algorithms because activity labels and models provide a rich vocabulary for expressing behavior within a system. In the past, theories about behavior and activities were formed based on limited observation. More recently, the maturing of technologies, such as the SHiB, has made it possible to automate activity learning. Learning activities in turn enriches smart homes because the home's intelligent agent can reason at a high level about the resident's activities and take appropriate actions.

In our building automation approach, activity learning plays two roles. First, activity recognition is used to identify activities as they are performed in a smart building environment. Second, activity prediction is used to forecast whether a particular activity will occur within the upcoming time window. Together, they provide a basis for building automation that supports current and upcoming tasks the residents will perform in the building. This section provides details for these two critical components of our activity-aware smart building.

The challenge of activity recognition is to map sensor events to a label that indicates the corresponding activity the individual is performing. There are activity recognition challenges that are unique among machine learning problems. The sequential nature of the input data, the ambiguous partitioning of data into activities, and the overlapping of activity classes mean that additional data processing must be performed. As Figure 6 shows, the recognition steps include collecting and preprocessing sensor data, dividing it into subsequences of manageable size, then extracting subsequence features. The final feature vectors are either labeled by an expert to use as training data or are input to an already-trained model to generate the corresponding activity label.



|  |  |  |  |
|---|---|---|---|
| 2009-06-08 | 13:48:49.033446 | M009 | ON |
| 2009-06-08 | 13:48:50.002373 | M023 | ON |
| 2009-06-08 | 13:48:51.042953 | M009 | OFF |
| 2009-06-08 | 13:48:52.024474 | M001 | ON |
| 2009-06-08 | 13:48:54.048006 | M016 | ON |
| 2009-06-08 | 13:48:55.015058 | M015 | ON |
| 2009-06-08 | 13:48:55.056504 | M008 | ON |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| M012 | M012 | M004 | M003 | M001 | M001 |  |  |
|  | M012 | M004 | M003 | M001 | M001 | M001 |  |
|  |  | M004 | M003 | M001 | M001 | M001 | D001 |

**Figure 6.** Activity recognition includes stages of raw sensor data collection, data preprocessing and segmentation, feature extraction, and supervised machine learning.

Let $A = \{a_1, a_2, .., a_T\}$ be the set of all modeled activities, where $a_i$ corresponds to the $i$th activity class. A smart home generates raw sensor data in the form of time-stamped sensor readings or events, $\Lambda = (\lambda_1, \lambda_2, .., \lambda_N)$, where event $\lambda_i$ corresponds to a sensor reading or sensor value generated at time $t_i$. The data are preprocessed to handle missing or noisy data, then, features $x \in \Re^d$ are extracted from the raw smart home sensor data. Finally, a supervised machine-learning algorithm learns a mapping from the feature vector $X$ to an activity label.

This work builds upon work by our team to design algorithms that automatically build activity models from sensor data using machine learning techniques [18–21]. Other groups have also explored a large number of approaches to supervised activity recognition [22–36]. These have been tested for a variety of sensor modalities, including environment [20,37–39], wearable [40–42], object [43,44], smart phones [45,46], and video [47]. The learning methods can be broadly categorized into template, generative, discriminative, and ensemble approaches. Template matching techniques employ a k nearest neighbor (kNN) classifier with dynamic time warping to a varying window size [48]. Generative approaches, such as naïve Bayes classifiers, Markov models and dynamic Bayes networks, have yielded promising results for behavior modeling and offline activity recognition when a large amount of labeled data is available [20,49–52]. On the other hand, discriminative approaches that model the boundary between different activity classes offer an effective alternative. These techniques include decision trees, meta classifiers based on boosting and bagging, support vector machines, and discriminative probabilistic graphical models such as conditional random fields [20,52–67]. Other approaches combine these underlying learning algorithms, including boosting and other ensemble methods [68–71].

The home automation approach described in this paper employs our activity recognition algorithm called CASAS-AR [18] to label raw data with corresponding activity labels. While many activity recognition algorithms have been proposed, they are typically designed for constrained situations with pre-segmented data, a single user, and no activity interruptions. CASAS-AR extends this to consider generalization of activity models over multiple smart homes. In earlier work, a common vocabulary of sensor locations was defined to facilitate the design of algorithms that recognize activities even in new environments with no training data. Furthermore, CASAS-AR provides real-time activity labeling on streaming data. To do this, CASAS-AR extracts features from a fixed-sized sliding window of sensor events, $\lambda_i \ldots \lambda_j$, and maps the feature vector onto an activity label, indicating the activity that was performed at the time of the last event in the window, or time $t_j$.

To train AR labels are provided for at least one month of sensor data from each smart building location. Human annotators label the sensor data in each dataset with corresponding activities based upon interviews with the residents, photographs of the home, and a floorplan highlighting the locations of sensors in the space. Sensor events are labeled with the activity that was determined to be occurring in the home at that time. In the experimental validation, 15 core activities are modeled that occur daily in a majority of the datasets. These activities are listed in Table 1.

Sensor events that do not fit into one of the core activity classes are labeled as "Other activity" and provide context for AR as well as for the activity forecaster. To maximize consistency of ground truth labels, multiple annotators look at the datasets and disagreements between labels are resolved via discussion. The annotators demonstrate inter-annotator agreement of $\kappa = 0.85$ for our selected activities. The approach is tested in ($n = 30$) smart homes with multi-year data, achieving >95% recognition accuracy based on 3-fold cross validation assessment with each sensor window treated as a separate data point.

Given the foundation of an activity recognition algorithm, such as CASAS-AR, the CPS can then perform activity prediction. In the context of building automation, activity prediction consists of determining which activities will occur within the next 10 min. Smart home-based activity prediction is a new area in the field and has not been used before this for home automation. This is the goal for building automation in this paper because the home can predict activities rather than just react to them, thereby making the home more efficient in its use of resources such as energy. Specifically, this approach allows the CARL building automation system to avoid turning off devices that are currently in use (as determined by activity recognition) or will soon be in use (determined by activity prediction). In contrast with activity recognition, the activity prediction problem is to determine whether a particular activity will occur within the next time window (here, the size of the time window is 10 min). This is viewed as a binary classification problem. As with activity recognition, the input consists of raw sensor events $\Lambda$, the duration of the prediction window, $w$, and a target activity, $a$. A feature vector $X'$ is extracted from the raw sensor data. Features are extracted for activity recognition (feature vector $X$) and for activity prediction (feature vector $X'$). A machine learning algorithm is then used to learn the mapping $h$: $< X', w, a > \rightarrow \{0,1\}$, from the input feature vector, window duration, and activity to a binary label where 0 indicates that activity $a$ will not occur in the next $w$ time units and 1 indicates that $a$ will occur.

**Table 1.** Activity classes.

| Activity | # Sensor Events |
|---|---|
| Bathe | 22,761 |
| Bed toilet transition | 6817 |
| Cook | 26,032 |
| Drink | 11,522 |
| Eat | 16,961 |
| Enter home | 1376 |
| Leave home | 2570 |
| Other activity | 791,938 |
| Relax | 8753 |
| Sleep | 793,531 |
| Toilet | 56,969 |
| Wash dishes | 2900 |
| Watch TV | 450,628 |
| Water plants | 2408 |
| Work on computer | 283,509 |

The activity prediction problem is formulated and solved in the framework of imitation learning. In traditional imitation learning, the goal of the learner is to learn to imitate the behavior of an expert performing a sequential decision making task (such as playing a game) in a way that generalizes to similar tasks or situations. Imitation learning techniques have been applied to a variety of natural language processing and computer vision prediction tasks [35,72,73]. In the activity prediction problem, the expert corresponds to a loss function $L$ and the expert behavior corresponds to predicting the best output at each time step. For each time step the activity prediction algorithm computes the feature vector and the correct activity prediction based on activity labels provided by CASAS-AR. If the algorithm can learn a function $h$ that is consistent with these imitation examples then the learned function will generalize and perform well on new instances [74,75].

In principle, any multi-output regression learner can be used for this task. However, inspired by the binary relevance (BR) classifier for multi-label classification [76], we decompose the multi-output regression problem by learning one regression function for each output variable (in this case, each activity's predicted next occurrence) independently. We have a hard learning problem at hand, which means linear functions will not suffice. We experimented with logistic regression, multi-layer perceptrons, and support vector machine regression. We also tested a standard regression tree, which is a decision tree which decomposes the regression space into subspaces based on values for attributes of the data instances. The specific attributes are chosen based on their ability to reduce the entropy of the data, then the output value is stored at the leaf nodes where the space cannot be further decomposed. The regression tree outperformed the other methods on average. However, like the other approaches, the regression tree could not handle the high variance for some of the activity times. Hence, we finally employed a variant of regression trees called model trees [77], where predictions are made by a learned linear function over all of the features at each leaf node of the decision tree. This performed consistently the best over the alternative regression methods and is used as the backbone of the CARL home automation algorithm.

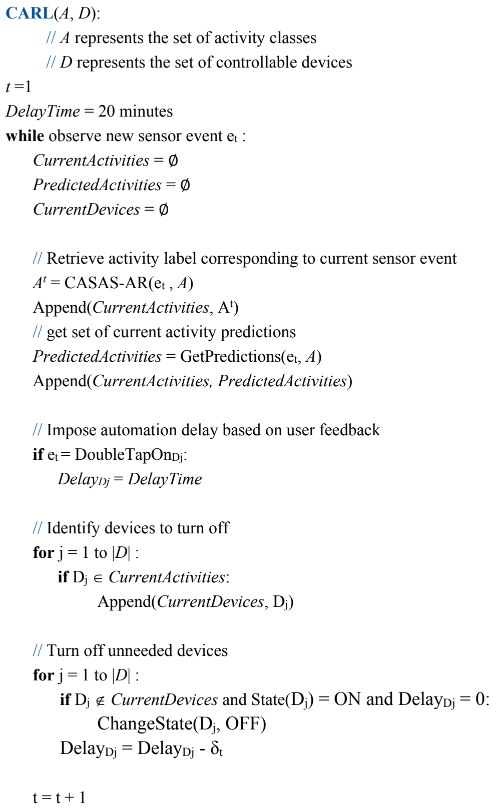
## 5. Activity-Aware Home Automation

CARL automates control of all devices in a smart building space using the smart building infrastructure described in Section 3 and the activity learning elements described in Section 4. The initial strategy of CARL is to turn off all devices that are not needed in support of the current set of activities as well as those that are anticipated to occur within the next 10 min.

As a preliminary step, CARL identifies the set of devices associated with each activity $a \in A$; these should not be turned off if $a$ is a current or upcoming activity. Here, $D_a$ represents devices that are associated with activity $a$, where $D$ is a subset of the total set of devices, $D_a \subseteq D$.

Assuming that the device sets have been constructed, CARL then performs a check for conditions to perform device automation at every time step $t$. The goal is to identify each device that is being used by a current or forthcoming activity, denoted as "*CurrentDevices*", and turn off every device not in the set "*CurrentDevices*". CARL must therefore identify the current activities being performed at time $t$ (*CurrentActivities*) and those that will occur between $t$ and $t$ + 10 min (*PredictedActivities*). The CASAS-AR activity recognizer is used to determine which activities are current at time $t$, or "*CurrentActivities*". CARL's GetPredictions function is used to determine which activities will occur within the next 10 min (between time $t$ and time $t$ + 10 min, or "*PredictedActivities*").

Finally, a command is sent to turn off the device. In some cases, the smart home resident may want the device remain on after CARL decided it is not needed. User buttons are installed around the home. A double tap on the button indicates that the user is overriding CARL to turn a device back on. A delay is imposed on the corresponding device called *DelayTime*, during which CARL will not turn off the device. For the experiments in this paper, *DelayTime* is set to twenty minutes. A summary of the CARL operations is given in Figure 7.



```
CARL($A$, $D$):
      // $A$ represents the set of activity classes
      // $D$ represents the set of controllable devices
$t$ =1
DelayTime = 20 minutes
while observe new sensor event $e_t$ :
    CurrentActivities = ∅
    PredictedActivities = ∅
    CurrentDevices = ∅

    // Retrieve activity label corresponding to current sensor event
    $A^t$ = CASAS-AR($e_t$ , $A$)
    Append(CurrentActivities, $A^t$)
    // get set of current activity predictions
    PredictedActivities = GetPredictions($e_t$, $A$)
    Append(CurrentActivities, PredictedActivities)

    // Impose automation delay based on user feedback
    if $e_t$ = DoubleTapOn$_{Dj}$:
        Delay$_{Dj}$ = DelayTime

    // Identify devices to turn off
    for j = 1 to |$D$| :
        if $D_j \in$ CurrentActivities:
              Append(CurrentDevices, $D_j$)

    // Turn off unneeded devices
    for j = 1 to |$D$| :
        if $D_j \notin$ CurrentDevices and State($D_j$) = ON and Delay$_{Dj}$ = 0:
              ChangeState($D_j$, OFF)
        Delay$_{Dj}$ = Delay$_{Dj}$ - $\delta_t$

    t = t + 1
```

**Figure 7.** CASAS activity aware resource learning (CARL) automation pseudocode.

## 6. Experimental Results

CARL's goal is to turn off as many as devices as possible without interfering with resident tasks. Performance can thus be measured in terms of the number of times a device is turned off (or the corresponding reduction in energy consumption) and the number of resident disruptions (or the number of times the user double tapped a switch in order to provide feedback to the system while turning the device back on.

Here, the CARL activity-aware automation architecture is validated using data collected from the Navan smart home described in Section 3. Because much of the system depends on the ability to correctly recognize activities in real time as they occur, first the accuracy of the CASAS-AR activity recognizer is evaluated on the Navan smart home data. Table 2 summarizes the performance of CASAS-AR using 3-fold cross validation. Table 2 also provides a confusion matrix that highlights where the errors lie. The performance evaluation is based on two months of smart home data, collected continuously while the resident performed normal routines. As can be seen in Table 2, the overall accuracy is high but the larger classes, such as "*Other activity*", create a class imbalance that introduces associated errors. CARL's automation effectiveness builds on this performance because CASAS-AR provides the set of current activities, *CurrentActivities*, used in Figure 8.

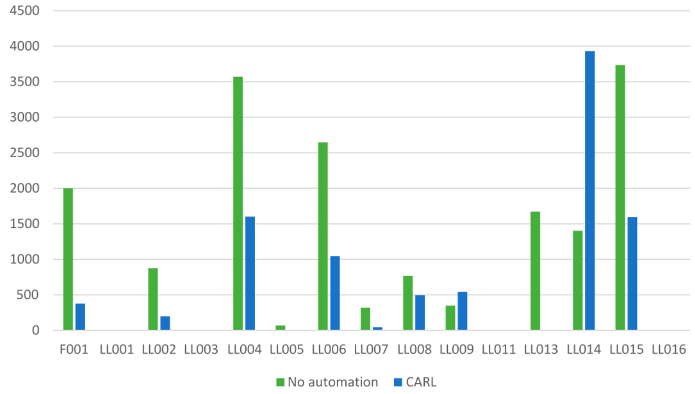


**Figure 8.** Minutes each device is on using the baseline method (no automation) and using CARL.

The next component is CARL's activity prediction. This is a binary classification problem, indicating for each activity whether it will occur in the next ten minutes (class = Yes) or not (class = No). While activity prediction in this case is expected to outperform activity recognition because there are fewer classes, this is not always the case. Table 3 summarizes the three-fold cross validation results of activity prediction for the Navan smart home. As can be seen in Table 3, activity prediction performance varies greatly between particular activities. Activities that occur often have enough training data to adequately learn the activity times. Activities that are highly predictable, such as sleep, also yield strong predictive accuracy. On the other hands, activities that are less predictable and less frequent have lower accuracy. An additional challenge is the extremely class imbalance in this learning problem. Most activities are not current much more than they are current. For any given activity it is expected that it will be current only $1.0/|A|$ of the time. Because there are 15 activities, the average time that an activity will occur is 0.067 of the time. Machine learning algorithms attempt to optimize classification accuracy. For imbalanced class distributions this means that most of the predictions will favor the majority class (the activity will not occur within the next 10 min) rather than the minority class (the activity will occur within the next 10 min). These influences are reflected in the results shown in Table 3.

**Table 2.** CASAS-AR performance on Navan smart home data. Accuracy #correctly classified activities/#total activities; G-mean = sqrt((true positive/(true positive + false negative)) × (true negative/(true negative + false positive))); Precision = true positive/(true positive + false positive); Recall = true positive/(true positive + false negative).

| Activity | Bathe | Bed Toilet | Cook | Drink | Eat | Enter Home | Leave Home | Other Activity | Relax | Sleep | Toilet | Wash Dishes | Watch TV | Water Plants | Work |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bathe | 1201 | 0 | 0 | 0 | 0 | 0 | 3 | 144 | 0 | 0 | 1059 | 0 | 0 | 0 | 11 |
| Bed toilet | 47 | 998 | 0 | 0 | 0 | 0 | 0 | 175 | 0 | 55 | 104 | 0 | 0 | 0 | 4 |
| Cook | 0 | 0 | 12 | 63 | 114 | 0 | 25 | 6358 | 0 | 0 | 0 | 0 | 76 | 0 | 9 |
| Drink | 0 | 0 | 35 | 80 | 0 | 0 | 0 | 2482 | 0 | 41 | 11 | 21 | 406 | 0 | 9 |
| Eat | 0 | 3 | 5 | 43 | 379 | 0 | 0 | 2873 | 0 | 9 | 29 | 0 | 41 | 0 | 0 |
| Enter home | 0 | 0 | 0 | 0 | 2 | 357 | 2 | 113 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Leave home | 0 | 0 | 3 | 1 | 11 | 21 | 207 | 336 | 0 | 2 | 0 | 0 | 23 | 0 | 2 |
| Other activity | 172 | 287 | 1865 | 2181 | 1574 | 93 | 486 | 88,216 | 48 | 863 | 852 | 58 | 5102 | 0 | 2686 |
| Relax | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 243 | 0 | 58 | 1 | 0 | 6 | 0 | 2 |
| Sleep | 0 | 61 | 0 | 1 | 9 | 0 | 0 | 874 | 4 | 8400 | 6 | 1 | 4 | 0 | 2 |
| Toilet | 1159 | 219 | 0 | 17 | 3 | 0 | 0 | 688 | 0 | 1 | 5667 | 0 | 148 | 0 | 22 |
| Wash dishes | 0 | 0 | 5 | 40 | 0 | 0 | 0 | 406 | 0 | 0 | 0 | 0 | 106 | 0 | 0 |
| Watch TV | 0 | 0 | 74 | 68 | 32 | 0 | 4 | 5347 | 51 | 1930 | 70 | 27 | 14,644 | 0 | 8 |
| Water plants | 0 | 0 | 1 | 84 | 0 | 0 | 0 | 428 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Work | 2 | 12 | 0 | 12 | 17 | 2 | 1 | 3206 | 0 | 2544 | 13 | 0 | 35 | 0 | 23,216 |
| Accuracy | 0.98 | 0.99 | 0.96 | 0.97 | 0.98 | 1.00 | 1.00 | 0.79 | 1.00 | 0.97 | 0.98 | 1.00 | 0.93 | 1.00 | 0.96 |
| Overall Accuracy | - | - | - | - | - | - | - | 0.74 | - | - | - | - | - | - | - |
| G Mean | - | - | - | - | - | - | - | 0.88 | - | - | - | - | - | - | - |
| Precision | - | - | - | - | - | - | - | 0.89 | - | - | - | - | - | - | - |
| Recall | - | - | - | - | - | - | - | 0.80 | - | - | - | - | - | - | - |

**Table 3.** Activity occurrence prediction performance on Navan smart home data.

| Performance Metric | Bathe | Bed Toilet | Cook | Drink | Eat | Enter Home | Leave Home | Other Activity | Relax | Sleep | Toilet | Wash Dishes | Watch TV | Water Plants | Work |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.93 | 0.89 | 0.99 | 0.89 | 0.98 | 0.94 | 0.96 | 0.71 | 1.00 | 0.89 | 0.75 | 0.99 | 0.87 | 1.00 | 0.78 |
| G Mean | 0.77 | 0.54 | 0.73 | 0.58 | 0.68 | 0.51 | 0.34 | 0.71 | 0.87 | 0.91 | 0.53 | 0.36 | 0.86 | 0.00 | 0.78 |
| Precision (False) | 0.99 | 0.99 | 1.00 | 0.98 | 1.00 | 0.99 | 0.99 | 0.82 | 1.00 | 0.99 | 0.90 | 1.00 | 0.94 | 1.00 | 0.95 |
| Precision (True) | 0.11 | 0.04 | 0.35 | 0.09 | 0.27 | 0.05 | 0.04 | 0.65 | 0.65 | 0.79 | 0.20 | 0.02 | 0.71 | 0.00 | 0.39 |
| Recall (False) | 0.93 | 0.90 | 0.99 | 0.90 | 0.99 | 0.95 | 0.97 | 0.59 | 1.00 | 0.83 | 0.80 | 0.99 | 0.87 | 1.00 | 0.78 |
| Recall (True) | 0.63 | 0.32 | 0.54 | 0.37 | 0.47 | 0.28 | 0.12 | 0.85 | 0.77 | 0.99 | 0.35 | 0.13 | 0.86 | 0.00 | 0.79 |

Finally, CARL is tested as a fully automated home control system in our Navan smart apartment. To do this, the CASAS-AR activity recognition algorithm and the CARL activity prediction algorithm are trained on three months of data with activity labels provided by a human annotator. The automation results are then collected for one week in the apartment. The training data and testing data were separated in time by several months, during which some routine changes would be expected due to concept draft, seasonal changes, and normal behavior variation.

Anecdotal information from the resident indicated that many of the activities were correctly detected, anticipated, and automated. However, the ones that were incorrect were often detrimental to the resident's comfort. In terms of quantifiable performance evaluation, two measures are used. The first is the number of "double button taps" performed by the resident. These represent false positive cases where CARL turned off a device at a time that was incorrect or inconvenient for the resident and the resident indicate the mistake by tapping the feedback button twice. The resident was at home almost the entire duration of the test week. However, during the times that he was out of the home he provided feedback by looking at the automation and sensor data logs to assess whether each automation step was appropriate or was incorrect.

Table 4 summarizes the performance of CARL in terms of its ability to accurately turn off devices when they are not needed. As the table indicates, not only does performance vary greatly from one device to another but it closely mirrors the activity recognition and activity prediction performance. As an example, Work is an activity with consistent recognition and prediction performance. Similarly, device LL015 is automated with strong true positive rates (TPR) and false negative rates (FNR). This indicates that as the ability of CASAS-AR and CARL's activity predictor improves, so will the ability to accurately automate home control. This can be accomplished through additional training data and greater consistency of human activity label annotations.

**Table 4.** Activity occurrence prediction performance on Navan smart home data by device. Device placement is shown in Figure 3. TRP: true positive rates; and FNR: false negative rates.

| Device | Automated Turn Off | Double Tap On | Manual Off | TPR | FNR |
|---|---|---|---|---|---|
| F001 | 12 | 2 | 3 | 0.83 | 0.80 |
| LL001 | 0 | 0 | 0 | 1.00 | 1.00 |
| LL002 | 6 | 0 | 13 | 1.00 | 0.32 |
| LL003 | 0 | 0 | 0 | 1.00 | 1.00 |
| LL004 | 18 | 3 | 5 | 0.83 | 0.78 |
| LL005 | 0 | 0 | 6 | 1.00 | 1.00 |
| LL006 | 28 | 17 | 4 | 0.39 | 0.88 |
| LL007 | 4 | 1 | 0 | 0.75 | 1.00 |
| LL008 | 29 | 10 | 9 | 0.66 | 0.76 |
| LL009 | 9 | 6 | 5 | 0.33 | 0.64 |
| LL011 | 0 | 0 | 1 | 1.00 | 0.00 |
| LL013 | 0 | 0 | 0 | 1.00 | 1.00 |
| LL014 | 41 | 31 | 12 | 0.24 | 0.77 |
| LL015 | 16 | 3 | 1 | 0.81 | 0.94 |
| LL016 | 0 | 0 | 0 | 1.00 | 1.00 |

Finally, Figures 8 and 9 show the minutes saved and energy reduced through CARL automation. Using activity-aware automation reduces device utilization by 56% and reduces energy consumption by 50%. Of course, this savings must be balanced with the 21% average positive rate. In some of these cases, the correct automation step was determined but was not executed at an optimal time. To analyze this type of error more carefully, was also compute the normalized root mean squared error (nRMSE) for each CARL-based device automation. Error is computed as the time between the device automation and when it should have been turned off based on the actual activities that occurred at each time step. Each error value is squared and the set of errors are normalized to fall within the range of 0–1. The nRMSE over the entire dataset is 0.138577. This indicates that CARL is able to automate devices based on its awareness of activities that are occurring in the home.

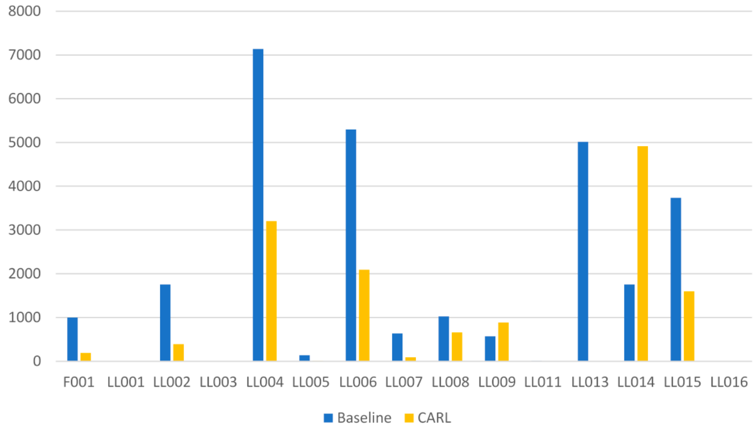


**Figure 9.** Energy consumed using the baseline method (no automation) and using CARL.

## 7. Conclusions

This paper introduces the notion of an activity-aware building automation system. Such a system can be used as a critical component of smart city automation to reduce energy consumption while supporting routine activities. Our activity-aware automation system, CARL, uses activity recognition to identify current activities and activity prediction to anticipate upcoming activities. Both sources of information are utilized to make decisions regarding devices to turn off, thus reducing energy consumption. Experiments run on an actual smart apartment indicate that CARL is aware of resident activities and is able to automate home devices based on this information in a way that reduces resource consumption.

There are many directions to consider for future work. A first step will be to use CARL to, not only turn off devices that are not currently needed in support of current activities, but to also automatically turn on devices that are needed. In addition, CARL's will be enhanced by segmenting and smoothing activities, reducing the amount of jitter in activity labels and improving activity prediction performance. Finally, CARL will be evaluated in a greater number of automated buildings to show the combined energy reduction that can be realized using an activity-aware approach to designing smart buildings and smart cities.

**Acknowledgments:** This work was supported in part by National Science Foundation Grant Nos. 1543656 and 1262814.

**Author Contributions:** Brian L. Thomas conceived the CASAS infrastructure, installed the Navan smart home testbed, implemented CARL, and performed the experiments. Diane J. Cook conceived the activity-aware cyber-physical system formalism and designed the activity recognition algorithms. Both authors contributed toward writing the paper.

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

## References

1. Cook, D.J.; Crandall, A.; Thomas, B.; Krishnan, N. CASAS: A smart home in a box. *IEEE Comput.* **2012**, *46*, 62–69. [CrossRef] [PubMed]
2. *April 2016 Monthly Energy Review*; U.S. Energy Information Administration (EIA): Washington, DC, USA, 2016.
3. *How Much Energy is Consumed in Residential and Commercial Buildings in the United States?* U.S. Energy Information Administration (EIA): Washington, DC, USA, 2016.
4. Darby, S.; Liddell, C.; Hills, D.; Drabble, D. *Smart Metering Early Learning Project: Synthesis Report*; Department of Energy & Climate Change: London, UK, 2015.

5. Allcott, H.; Rogers, T. The short-run and long-run effects of behavioral interventions: Experimental evidence from energy conservation. *Am. Econ. Rev.* **2014**, *104*, 3003–3037. [CrossRef]
6. Szewcyzk, S.; Dwan, K.; Minor, B.; Swedlove, B.; Cook, D.J. Annotating smart environment sensor data for activity learning. *Technol. Health Care* **2009**, *17*, 161–169. [PubMed]
7. Wilson, C.; Dowlatabadi, H. Models of decision making and residential energy use. *Annu. Rev. Environ. Resour.* **2007**, *32*, 169–203. [CrossRef]
8. Brounen, D.; Kok, N.; Quigley, J.M. Residential energy use and conservation: Economics and demographics. *Eur. Econ. Rev.* **2012**, *56*, 931–945. [CrossRef]
9. Darby, S. Smart metering: What potential for householder engagement? *Build. Res. Inf.* **2010**, *38*, 442–457. [CrossRef]
10. Riche, Y.; Dodge, J.; Metoyer, R. Studying always-on electricity feedback in the home. In Proceedings of the International Conference on Human Factors in Computing Systems, Atlanta, GA, USA, 10–15 April 2010; pp. 1995–1998.
11. Kim, C.G.; Kim, K.J. Implementation of a cost-effective home lighting control system on embedded Linux with OpenWrt. *Pers. Ubiquitous Comput.* **2014**, *18*, 535–542. [CrossRef]
12. *Speech by the Rt Hon Patricia Hewitt MP, Secretary of State for Health. in Long-term Conditions Alliance Annual Conference*; Department of Health: Providence, RI, USA, 2007.
13. Kidd, C.D.; Orr, R.; Abowd, G.D.; Atkeson, C.G.; Essa, I.A.; MacIntyre, B.; Mynatt, E.D.; Starner, T.; Newstetter, W. The aware home: A living laboratory for ubiquitous computing research. In Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture—CoBuild'99, Pittsburgh, PA, USA, 1–2 October 1999.
14. Helal, S.; Mann, W.; El-Zabadani, H.; King, J.; Kaddoura, Y.; Jansen, E. The gator tech smart house: A programmable pervasive space. *Computer* **2005**, *38*, 50–60. [CrossRef]
15. Cook, D.J.; Youngblood, M.; Heierman, E.O.; Gopalratnam, K.; Rao, S.; Litvin, A.; Khawaja, F. MavHome: An agent-based smart home. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, Fort Worth, TX, USA, 26 March 2003; pp. 521–524.
16. Elfaham, A.; Hagras, H.; Helal, S.; Hossain, S.; Lee, J.W.; Cook, D. A fuzzy based verification agent for the Persim human activity simulator in ambient intelligent environments. In Proceedings of the 2010 IEEE International Conference on Fuzzy Systems, Barcelona, Spain, 18–23 July 2010.
17. Cook, D.J.; Das, S. Pervasive computing at scale: Transforming the state of the art. *Pervasive Mob. Comput.* **2012**, *8*, 22–35. [CrossRef]
18. Krishnan, N.; Cook, D.J. Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* **2014**, *10*, 138–154. [CrossRef] [PubMed]
19. Cook, D.J.; Krishnan, N.; Rashidi, P. Activity discovery and activity recognition: A new partnership. *IEEE Trans. Syst. Man Cybern. Part B* **2013**, *43*, 820–828. [CrossRef] [PubMed]
20. Cook, D.J. Learning setting-generalized activity models for smart spaces. *IEEE Intell. Syst.* **2012**, *27*, 32–38. [CrossRef] [PubMed]
21. Crandall, A.; Cook, D.J. *Human Aspects in Ambient Intelligence*; Atlantis Press: Paris, France, 2013; pp. 55–71.
22. Aggarwal, J.K.; Ryoo, M.S. Human activity analysis: A review. *ACM Comput. Surv.* **2011**, *43*. [CrossRef]
23. Chen, L.; Hoey, J.; Nugent, C.D.; Cook, D.J.; Yu, Z. Sensor-based activity recognition. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 790–808. [CrossRef]
24. Ke, S.R.; Thuc, H.L.U.; Lee, Y.J.; Hwang, J.N.; Yoo, J.H.; Choi, K.H. A review on video-based human activity recognition. *Computers* **2013**, *2*, 88–131. [CrossRef]
25. Bulling, A.; Blanke, U.; Schiele, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.* **2014**, *46*, 107–140. [CrossRef]
26. Reiss, A.; Stricker, D.; Hendeby, G. Towards robust activity recognition for everyday life: Methods and evaluation. In Proceedings of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, Venice, Italy, 5–8 May 2013; pp. 25–32.
27. Vishwakarma, S.; Agrawal, A. A survey on activity recognition and behavior understanding in video surveillance. *Vis. Comput.* **2013**, *29*, 983–1009. [CrossRef]
28. Lara, O.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1192–1209. [CrossRef]

29. Chen, L.; Khalil, I. *Activity Recognition in Pervasive Intelligent Environments*; Chen, L., Nugent, C.D., Biswas, J., Hoey, J., Eds.; Atlantis Press: Paris, France, 2011; pp. 1–31.
30. Tuaraga, P.; Chellappa, R.; Subrahmanian, V.S.; Udrea, O.; Turaga, P. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1473–1488. [CrossRef]
31. Alon, J.; Athitsos, V.; Yuan, Q.; Sclaroff, S. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 1685–1699. [CrossRef] [PubMed]
32. Iglesias, J.A.; Angelov, P.; Ledezma, A.; Sanchis, A. Human activity recognition based on evolving fuzzy systems. *Int. J. Neural Syst.* **2010**, *20*, 355–364. [CrossRef] [PubMed]
33. Liao, I.L.; Fox, D.; Kautz, H. Location-based activity recognition using relational Markov networks. In Proceedings of the International Joint Conference on Artificial Intelligence, Edinburgh, UK, 30 July–5 August 2005; pp. 773–778.
34. Guenterberg, E.; Ghasemzadeh, H.; Jafari, R. Automatic segmentation and recognition in body sensor networks using a hidden Markov model. *ACM Trans. Embed. Comput. Syst.* **2012**, *11*. [CrossRef]
35. Doppa, J.R.; Fern, A.; Tadepalli, P. Structured prediction via output space search. *J. Mach. Learn. Res.* **2014**, *15*, 1317–1350.
36. Doppa, J.R.; Fern, A.; Tadepalli, P. HC-Search: Learning heuristics and cost functions for structured prediction. *J. Artif. Intell. Res.* **2014**, *50*, 369–407.
37. Hagras, H.; Doctor, F.; Lopez, A.; Callaghan, V. An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 41–55. [CrossRef]
38. Munguia-Tapia, E.; Intille, S.S.; Larson, K. Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Comput.* **2004**, *3001*, 158–175.
39. Wan, J.; O'Grady, M.J.; O'Hare, G.M. Dynamic sensor event segmentation for real-time activity recognition in a smart home context. *Pers. Ubiquitous Comput.* **2015**, *19*, 287–301. [CrossRef]
40. Jarafi, R.; Sastry, S.; Bajcsy, R. Distributed recognition of human actions using wearable motion sensor networks. *J. Ambient Intell. Smart Environ.* **2009**, *1*, 103–115.
41. Junker, H.; Amft, O.; Lukowicz, P.; Groster, G. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognit.* **2008**, *41*, 2010–2024. [CrossRef]
42. Mukhopadhyay, S.C. Wearable sensors for human activity monitoring: A review. *IEEE Sens. J.* **2014**, *15*, 1321–1330. [CrossRef]
43. Gu, T.; Chen, S.; Tao, X.; Lu, J. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data Knowl. Eng.* **2010**, *69*, 533–544. [CrossRef]
44. Philipose, M.; Fishkin, K.P.; Perkowitz, M.; Patterson, D.J.; Fox, D.; Kautz, H.; Hahnel, D. Inferring activities from interactions with objects. *IEEE Pervasive Comput.* **2004**, *3*, 50–57. [CrossRef]
45. Gyorbiro, N.; Fabian, A.; Homanyi, G. An activity recognition system for mobile phones. *Mob. Netw. Appl.* **2008**, *14*, 82–91. [CrossRef]
46. Kwapisz, J.R.; Weiss, G.M.; Moore, S.A. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explor. Newsl.* **2010**, *12*, 74–82. [CrossRef]
47. Candamo, J.; Shreve, M.; Goldgof, D.; Sapper, D.; Kasturi, R. Understanding transit scenes: A survey on human behavior recognition algorithms. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 206–224. [CrossRef]
48. Forster, K.; Monteleone, S.; Calatroni, A.; Roggen, D.; Troster, G. Incremental kNN classifier exploiting correct-error teacher for activity recognition. In Proceedings of the 2010 Ninth International Conference on Machine Learning and Applications, Washington, DC, USA, 12–14 December 2010; pp. 445–450.
49. Amft, O.; Troster, G. On-body sensing solutions for automatic dietary monitoring. *IEEE Pervasive Comput.* **2009**, *8*, 62–70. [CrossRef]
50. Zhang, M.; Sawchuk, A.A. Motion primitive-based human activity recognition using a bag-of-features approach. In Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, Miami, FL, USA, 28–30 January 2012; pp. 631–640.
51. Abdullah, S.; Lane, N.D.; Choudhury, T. Towards population scale activity recognition: A framework for handling data diversity. In Proceedings of the National Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012.

52. Hirano, T.; Maekawa, T. A hybrid unsupervised/supervised model for group activity recognition. In Proceedings of the International Symposium on Wearable Computers, Zurich, Switzerland, 9–12 September 2013; pp. 21–24.
53. Hung, H.; Englebienne, G.; Kools, J. Classifying social actions with a single accelerometer. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 8–12 September 2013; pp. 207–210.
54. Petersen, J.; Larimer, N.; Kaye, J.A.; Pavel, M.; Hayes, T.L. SVM to detect the presence of visitors in a smart home environment. In Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society, San Diego, CA, USA, 28 August–1 September 2012; pp. 5850–5853.
55. Kjaergaard, M.B. Studying sensing-based systems: Scaling to human crowds in the real world. *IEEE Comput.* **2013**, *17*, 80–84. [CrossRef]
56. Kjaergaard, M.B.; Wirz, M.; Roggen, D.; Troster, G. Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones. In Proceedings of the ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 240–249.
57. Gordon, D.; Hanne, J.-H.; Berchtold, M.; Shirehjini, A.A.N.; Beigl, M. Towards collaborative group activity recognition using mobile devices. *Mob. Netw. Appl.* **2013**, *18*, 326–340. [CrossRef]
58. Lu, C.H.; Chiang, Y.T. Interaction-enabled multi-user model learning for a home environment using ambient sensors. *IEEE J. Biomed. Health Inf.* **2015**, *29*, 1015–1046.
59. Wang, L.; Gu, T.; Tao, X.; Chen, H.; Lu, J. Multi-user activity recognition in a smart home. *Atl. Ambient Pervasive Intell.* **2011**, *4*, 59–81.
60. Wu, T.; Lian, C.; Hsu, J.Y. Joint recognition of multiple concurrent activities using factorial conditional random fields. In Proceedings of the Association for the Advancement of Artificial Intelligence Workshop on Plan, Activity, and Intent Recognition, Palo Alto, CA, USA, 28–29 June 2007.
61. Tolstikov, A.; Phus, C.; Biswas, J.; Huang, W. Multiple people activity recognition using MHT over DBN. In Proceedings of the 9th International Conference on Smart Homes and Health Telematics, Montreal, QC, Canada, 12–15 June 2011; pp. 313–318.
62. Hu, D.H.; Yang, Q. CIGAR: Concurrent and interleaving goal and activity recognition. In Proceedings of the 23rd National Conference on Artificial Intelligence, Chicago, IL, USA, 13–17 July 2008; pp. 1363–1368.
63. Chiang, Y.T.; Hsu, K.C.; Lu, C.H.; Fu, L.C. Interaction models for multiple-resident activity recognition in a smart home. In Proceedings of the International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3753–3758.
64. Gu, T.; Wang, L.; Chen, H.; Tao, X.; Lu, J. Recognizing multiuser activities using wireless body sensor networks. *IEEE Trans. Mob. Comput.* **2011**, *10*, 1618–1631. [CrossRef]
65. Blanke, U.; Schiele, B.; Kreil, M.; Lukowicz, P.; Sick, B.; Gruber, T. All for one or one for all? Combining heterogeneous features for activity spotting. In Proceedings of the 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, Mannheim, Germany, 29 March–2 April 2010; pp. 18–24.
66. Van Kasteren, T.; Noulas, A.; Englebienne, G.; Krose, B. Accurate activity recognition in a home setting. In Proceedings of the ACM Conference on Ubiquitous Computing, Seoul, Korea, 21–24 September 2008.
67. Bulling, A.; Ward, J.A.; Gellersen, H. Multimodal recognition of reading activity in transit using body-worn sensors. *ACM Trans. Appl. Percept.* **2012**, *9*. [CrossRef]
68. Wang, S.; Pentney, W.; Popescu, A.M.; Choudhury, T.; Philipose, M. Common sense based joint training of human activity recognizers. In Proceedings of the International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007; pp. 2237–2242.
69. Lester, J.; Choudhury, T.; Borriello, G. A practical approach to recognizing physical activities. In Proceedings of the International Conference on Pervasive Computing, Sydney, Austrilia, 14–18 March 2006.
70. Hong, J.H.; Ramos, J.; Dey, A.K. Toward personalized activity recognition systems with a semipopulation approach. *IEEE Trans. Hum. Mach. Syst.* **2016**, *46*, 101–112. [CrossRef]
71. Lichman, M.; Smyth, P. Modeling human location data with mixtures of kernel densities. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014.

72. Ma, C.; Doppa, J.R.; Orr, J.W.; Mannem, P.; Fern, X.Z.; Dietterich, T.G.; Tadepalli, P. Prune-and-score: Learning for greedy coreference resolution. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
73. Xie, J.; Ma, C.; Doppa, J.R.; Mannem, P.; Fern, X.; Dietterich, T.; Tadepalli, P. Learning greedy policies for the easy-first framework. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
74. Khardon, R. Learning to take actions. *Mach. Learn. J.* **1999**, *35*, 57–90. [CrossRef]
75. Ross, S.; Gordon, G.J.; Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. *J. Mach. Learn. Res.* **2011**, *15*, 627–635.
76. Doppa, J.R.; Yu, J.; Ma, C.; Fern, A.; Tadepalli, P. HC-Search for multi-label prediction: An empirical study. In Proceedings of the National Conference on Artificial Intelligence, Quebec, QC, Canada, 27–31 July 2014.
77. Landwehr, N.; Hall, M.; Frank, E. Logistic model trees. In Proceedings of the European Conference on Machine Learning, Dubrovnik, Croatia, 22–26 September 2003; pp. 241–252.