

# A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning

Damien Bouchabou <sup>1,2,\*</sup> , Sao Mai Nguyen <sup>1\*</sup> , Christophe Lohr <sup>1</sup> , Benoit LeDuc <sup>2</sup> and Ioannis Kanellos <sup>1</sup> 

<sup>1</sup> IMT Atlantique Engineer School, Brest 29238, France; damien.bouchabou@imt-atlantique.fr (D.B); nguyensmai@gmail.com (S.M.N.); christophe.lohr@imt-atlantique.fr (C.L.); ioannis.kanellos@imt-atlantique.fr (I.K.)

<sup>2</sup> Delta Dore Company, Bonnemain 35270, France; bleduc@deltadore.com

\* Correspondence: damien.bouchabou@imt-atlantique.fr; nguyensmai@gmail.com

**Abstract:** Recent advances in Internet of Things (IoT) technologies and the reduction in the cost of sensors have encouraged the development of smart environments, such as smart homes. Smart homes can offer home assistance services to improve the quality of life, autonomy, and health of their residents, especially for the elderly and dependent. To provide such services, a smart home must be able to understand the daily activities of its residents. Techniques for recognizing human activity in smart homes are advancing daily. However, new challenges are emerging every day. In this paper, we present recent algorithms, works, challenges, and taxonomy of the field of human activity recognition in a smart home through ambient sensors. Moreover, since activity recognition in smart homes is a young field, we raise specific problems, as well as missing and needed contributions. However, we also propose directions, research opportunities, and solutions to accelerate advances in this field.

**Keywords:** survey; human activity recognition; deep learning; smart home; ambient assisting living; taxonomies; challenges; opportunities

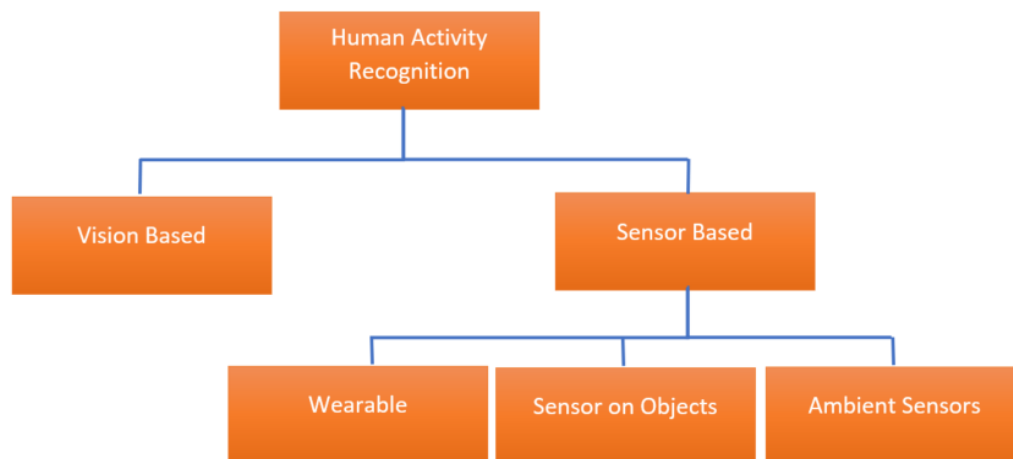
---

## 1. Introduction

With an aging population, providing automated services to enable people to live as independently and healthily as possible in their own homes has opened up a new field of economics [1]. Thanks to advances in the Internet of Things (IoT), the smart home is the solution being explored today to provide home services, such as health care monitoring, assistance in daily tasks, energy management, or security. A smart home is a house equipped with many sensors and actuators that can detect the opening of doors, the luminosity of the rooms, their temperature and humidity, etc. However, also to control some equipment of our daily life, such as heating, shutters, lights, or our household appliances. More and more of these devices are now connected and controllable at a distance. It is now possible to find in the houses, televisions, refrigerators, and washing machines known as intelligent, which contain sensors and are controllable remotely. All these devices, sensors, actuators, and objects can be interconnected through communication protocols.

In order to provide all of these services, a smart home must understand and recognize the activities of residents. To do so, the researchers are developing the techniques of Human Activity Recognition (HAR), which consists of monitoring and analyzing the behavior of one or more people to deduce the activity that is carried out. The various systems for HAR [2] can be divided into two categories [3]: video-based systems and sensor-based systems (see Figure 1).

**Citation:** Bouchabou, D.; Nguyen, S.M.; Lohr, C.; LeDuc, B.; Kanellos, I. A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning. *Sensors* **2021**, *21*, 6037. <https://doi.org/10.3390/s21186037>



**Figure 1.** Human Activity Recognition approaches.

### 1.1. Vision-Based

The vision-based HAR uses cameras to track human behavior and changes in the environment. This approach uses computer vision techniques, e.g., marker extraction, structure model, motion segmentation, action extraction, and motion tracking. Researchers use a wide variety of cameras, from simple RGB cameras to more complex systems by fusion of several cameras for stereo vision or depth cameras able to detect the depth of a scene with infrared lights. Several survey papers about vision-based activity recognition have been published [3,4]. Beddiar et al. [4] aims to provide an up-to-date analysis of vision-based HAR-related literature and recent progress.

However, these systems pose the question of acceptability. A recent study [5] shows that the acceptability of these systems depends on users' perception of the benefits that such a smart home can provide. It also conditions their concerns about the monitoring and sharing the data collected. This study shows that older adults (ages 36 to 70) are more open to tracking and sharing data, especially if it is useful to their doctors and caregivers, while younger adults (up to age 35) are rather reluctant to share information. This observation argues for less intrusive systems, such as smart homes based on IoT sensors.

### 1.2. Sensor-Based

HAR from sensors consists of using a network of sensors and connected devices, to track a person's activity. They produce data in the form of a time series of state changes or parameter values. The wide range of sensors—contact detectors, RFID, accelerometers, motion sensors, noise sensors, radar, etc.—can be placed directly on a person, on objects or in the environment. Thus, the sensor-based solutions can be divided into three categories, respectively: Wearable [6], Sensor on Objects [7], and Ambient Sensor [8].

Considering the privacy issues of installing cameras in our personal space, to be less intrusive and more accepted, sensor-based systems have dominated the applications of monitoring our daily activities [2,9]. Owing to the development of smart devices and Internet of Things, and the reduction of their prices, ambient sensor-based smart homes have become a viable technical solution which now needs to find human activity algorithms to uncover their potential.

### 1.3. Key Contributions

While existing surveys [2,10–13] report past works in sensor-based HAR in general, we will focus in this survey on algorithms for human activity recognition in smart homes and its particular taxonomies and challenges for the ambient sensors, which we will develop in the next sections. Indeed, HAR in smart homes is a challenging problem because the human activity is something complex and variable from a resident to another. Every resident has different lifestyles, habits, or abilities. The wide range of daily activities, as well as

the variability and the flexibility in how they can be performed, requires an approach that is scalable and must be adaptive.

Many methods have been used for the recognition of human activity. However, this field still faces many technical challenges. Some of these challenges are common to other areas of pattern recognition (Section 2) and, more recently, on automatic features extraction algorithms (Section 3), such as computer vision and natural language processing, while some are specific to sensor-based activity recognition, and some are even more specific to the smart home domain. This field requires specific methods for real-life applications. The data have a specific temporal structure (Section 4) that needs to be tackled, as well as poses challenges in terms of data variability (Section 5) and availability of datasets (Section 6) but also specific evaluation methods (Section 7). The challenges are summarized in Figure 2.

To carry out our review of the state of the art, we searched the literature for the latest advances in the field. We took the time to reproduce some works to confirm the results of works proposing high classification scores. In this study, we were able to study and reproduce the work of Reference [14–16], which allowed us to obtain a better understanding of the difficulties, challenges, and opportunities in the field of HAR in smart homes.

Compared with existing surveys, the key contributions of this work can be summarized as follows:

- We conduct a comprehensive survey of recent methods and approaches for human activity recognition in smart homes.
- We propose a new taxonomy of human activity recognition in smart homes in the view of challenges.
- We summarize recent works that apply deep learning techniques for human activity recognition in smart homes.
- We discuss some open issues in this field and point out potential future research directions.



Figure 2. Challenges for Human Activity Recognition in smart homes.

## 2. Pattern Classification

Algorithms for Human Activity Recognition (HAR) in smart homes are first pattern recognition algorithms. The methods found in the literature can be divided into two broad categories: Data-Driven Approaches (DDA) and Knowledge-Driven Approaches (KDA). These two approaches are opposite. DDA uses user-generated data to model and recognize the activity. They are based on data mining and machine learning techniques. KDA uses expert knowledge and rule design. They use prior knowledge of the domain, its modeling, and logical reasoning.

### 2.1. Knowledge-Driven Approaches (KDA)

In KDA methods, an activity model is built through the incorporation of rich prior knowledge gleaned from the application domain, using knowledge engineering and knowledge management techniques.

KDA are motivated by real-world observations that involve activities of daily living and lists of objects required for performing such activities. In real life situations, even if the activity is performed in different ways, the number and objects type involved do not vary significantly. For example, the activity “brush teeth” contain actions involving a toothbrush, toothpaste, water tap, cup, and towel. On the other hand, as humans have different lifestyles, habits, and abilities, they can perform various activities in different ways. For instance, the activity “make coffee” could be very different from one person to another.

KDA are founded upon the observations that most activities, specifically, routine activities of daily living and working, take place in a relatively circumstance of time, location, and space. For example, brushing teeth is normally undertaken twice a day in a bathroom in the morning and before going to bed and involves the use of toothpaste and toothbrush. Thin implicit relationships between activities, related temporal and spatial context and the entities involved, provide a diversity of hints and heuristics for inferring activities.

The knowledge structure is modeled and represented through forms, such as schemas, rules, or networks. KDA modeling and recognition intends to make use of rich domain knowledge and heuristics for activity modeling and pattern recognition. Three sub-approaches exist to use KDA: mining-based approach [17], logic-based approach [18], and ontology-based approach.

Ontology-based approaches are the most commonly used, as ontological activity models do not depend on algorithmic choices. They have been utilized to construct reliable activity models. Chen et al., in Reference [19], proposed an overview. Yamada et al. [20] use ontologies to represent objects in an activity space. Their work exploits the semantic relationship between objects and activities. A teapot is used in an activity of tea preparation, for example. This approach can automatically detect possible activities related to an object. It can also link an object to several representations or variability of an activity.

Chen et al. [21–23] constructed context and activity ontologies for explicit domain modeling.

KDA have the advantages to formalize activities and propose semantic and logical approaches. Moreover, these representations try to be the most complete as possible to overcome the activity diversity. However, the limitations of these approaches are the complete domain knowledge requirements to build activities models and the weakness in handling uncertainty and adaptability to changes and new settings. They need domain experts to design knowledge and rules. New rules can break or bypass the previous rules. These limitations are partially solved in the DDA approaches.

### 2.2. Data-Driven Approaches (DDA)

The DDA for HAR include both supervised and unsupervised learning methods, which primarily use probabilistic and statistical reasoning. Supervised learning requires labeled data on which an algorithm is trained. After training, the algorithm is then able to classify the unknown data.

The DDA strength is the probabilistic modeling capacity. These models are capable of handling noisy, uncertain, and incomplete sensor data. They can capture domain heuristics, e.g., some activities are more likely than others. They do not require a predefined domain knowledge. However, DDA require much data and, in the case of supervised learning, clean and correctly labeled data.

We observe that decision trees [24], conditional random fields [25], or support vector machines [26] have been used for HAR. Probabilistic classifiers, such as the Naive Bayes classifier [27–29], also showed good performance in learning and classifying offline activities when a large amount of training data is available. Sedkly et al. [30] evaluated several classification algorithms, such as AdaBoost, Cortical Learning Algorithm (CLA), Decision Trees, Hidden Markov Model (HMM), Multi-layer Perceptron (MLP), Structured Perceptron, and Support Vector Machines (SVM). They reported superior performance of DT, LSTM, SVM and stochastic gradient descent of linear SVM, logistic regression, or regression functions.

### 2.3. Outlines

To summarize, KDA propose to model activities following expert engineering knowledge, which is time consuming and difficult to maintain in case of evolution. DDA seems to yield good recognition levels and promises to be more adaptive to evolution and new situations. However, the DDA only yield good performance when given well-designed features as inputs. DDA needs more data and computation time than KDA, but the increasing number of datasets and the increasing computation power minimizes these difficulties and allows, today, even more complex models to be trained, such as Deep Learning(DL) models, which can overcome the dependency on input features.

## 3. Features Extraction

While the most promising algorithms for Human Activity Recognition in smart homes seem to be machine learning techniques, we describe how their performance depends on the features used as input. We describe how more recent machine learning has tackled this issue to generate automatically these features, as well as to propose end-to-end learning. We then highlight an opportunity to generate these features, while taking advantage of the semantic of human activity.

### 3.1. Handcrafted Features

In order to recognize the activities of daily life in smart homes, researchers first used manual methods. These handcrafted features are made after segmentation of the dataset into explicit activity sequences or windows. In order to provide efficient activity recognition systems, researchers have studied different features [31].

Initially, Krishann et al. [32] and Yala et al. [33] proposed several feature vector extraction methods described below: baseline, time dependency, sensor dependency, and sensor dependency extension. These features are then used by classification algorithms, such as SVM or Random Forest, to perform the final classification.

Inspired by previous work, more recently, Aminikhanghahi et al. [34] evaluate different types of sensor flow segmentations. However, they also listed different handmade features. Temporal features, such as day of the week, time of day, number of seconds since midnight, or time between sensor transitions, have been studied. Spatial features were also evaluated, such as location. However, metrics, such as the number of events in the window or the identifier of the sensor, also appear most frequently in the previous segments.

#### 3.1.1. The Baseline Method

This consists of extracting a feature vector from each window. It contains the time of the first and last sensor events in the window, the duration of the window, and a simple count of the different sensor events within the window. The size of the feature vector depends on the number of sensors in the datasets. For instance, if the dataset contains

34 sensors, the vector size will be  $34 + 3$ . From this baseline, researchers upgrade the method to overcome different problems or challenges.

### 3.1.2. The Time Dependence Method

This tries to overcome the problem of the sampling rate of sensor events. In most dataset, sensor events are not sampled regularly, and the temporal distance of an event from the last event in the segment has to be taken into account. To do this, the sensors are weighted according to their temporal distance. The more distant the sensor is in time, the less important it is.

### 3.1.3. The Sensor Dependency Method

This has been proposed to address the problem of the relationship between the events in the segment. The idea is to weight the sensor events in relation to the last sensor event in the segment. The weights are based on a matrix of mutual information between sensors, calculated offline. If the sensor appears in pair with the last sensor of the segment in other parts of the sensor flow, then the weight is high, and respectably low otherwise.

### 3.1.4. The Sensor Dependency Extension Method

This proposes to add the frequency of the sensor pair in the mutual information matrix. The more frequently a pair of sensors appears together in the dataset, the greater their weight.

### 3.1.5. The Past Contextual Information Method

This is an extension of the previous approaches to take into account information from past sessions. The classifier does not know the activity of the previous segment. For example, the activity “enter home” can only appear after the activity “leave home”. Naively the previous activity cannot be added to the feature vector. The algorithm might not be able to generalize enough. Therefore, Krishnan et al. [32] propose a two-part learning process. First, the model is trained without knowing the previous activity. Then, each prediction of activity in the previous segment is given to the classifier when classifying the current segment.

### 3.1.6. The Latent Knowledge Method

This was recently proposed by Surong et al. [16]. They improved these features by adding probability features. These additional features are learned from explicit activity sequences, in an unsupervised manner by a HMM and a Bayesian network. In their work, Surong et al. compared these new features with features extracted by deep learning algorithms, such as LSTM and CNN. The results obtained with these unsupervised augmented features are comparable to deep learning algorithms. They conclude that unsupervised learning significantly improves the performance of handcrafted features.

## 3.2. Automatic Features

In the aforementioned works, machine learning methods for the recognition of human activity make use of handcrafted features. However, these extracted features are carefully designed and heuristic. There is no universal or systematic approach for feature extraction to effectively capture the distinctive features of human activities.

Cook et al. [32] introduced, few years ago, an unsupervised method of discovering activities from sensor data based on a traditional machine learning algorithm. The algorithm searches for a sequence pattern that best compresses the input dataset. After many iterations, it reports the best patterns. These patterns are then clustered and given to a classifier to perform the final classification.

In recent years, deep learning has flourished remarkably by modeling high-level abstractions from complex data [35] in many fields, such as computer vision, natural language processing, and speech processing [6]. Deep learning models have the end-

to-end learning capability to automatically learn high-level features from raw signals without the guidance of human experts, which facilitates their wide applications. Thus, researchers used Multi-layer Perceptron (MLP) in order to carry out the classification of the activities [36,37]. However, the key point of deep learning algorithms is their ability to learn features directly from the raw data in a hierarchical manner, eliminating the problem of crafty approximations of features. They can also perform the classification task directly from their own features. Wang et al. [12] presented a large study on deep learning techniques applied to HAR with the sensor-based approach. Here, only the methods applied to smart homes are discussed.

### 3.2.1. Convolutional Neural Networks (CNN)

Works using Convolutional Neural Networks (CNN) have been carried out by the researchers. The CNN have demonstrated their strong capacity to extract characteristics in the field of image processing and time series. The CNN have two advantages for the HAR. First, they can capture local dependency, i.e., the importance of nearby observations correlated with the current event. In addition, they are scale invariant in terms of step difference or event frequency. Moreover, they are able to learn a hierarchical representation of the data. There are two types of CNN: 2D CNN for image processing and 1D CNN for sequence processing.

Gochoo et al. [15] have transformed activity sequences into binary images in order to use 2D CNN-based structures. Their work showed that this type of structure could be applied to the HAR. In an extension, G [redacted] et al. [38] propose to use colored pixels in the image to encode new sensor information about the activity in the image. Their extension proposes a method to encode sensors, such as temperature sensors, which are not binary, as well as the link between the different segments. Mohamed et al. [39] adopt the same strategy but convert activities into grayscale images. The gray value is correlated to the duration of sensor activation. The AlexNet structure [40] is then used for the feature extraction part of the images. Then, these features are used with classifiers to recognize the final activity.

Singh et al. [41] use a CNN 1D-based structure on raw data sequences for their high feature extraction capability. Their experiments show that the CNN 1D architecture achieves similar height results.

### 3.2.2. Autoencoder Method

Autoencoder is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible. Autoencoder, by design, reduces data dimensions by learning how to ignore the noise in the data. Researchers have explored this possibility because of the strong capacity of Autoencoders to generate the most discriminating features. The reduced encoded representation created by the Autoencoder contains the features that allow for discrimination of the activities.

Wang et al., in Reference [42], apply a two-layer Stacked Denoising Autoencoder (SDAE) to automatically extract unsupervised meaningfully features. The input of the SDAE are feature vectors extracted from 6-s time windows without overlap. The feature vector size is the number of sensors in the dataset. They compared two features forms: binary representation and numerical representation. The numerical representation method records the number of firing of a sensor during the time window, while the binary representation method sets to one the sensor value if this one fired in the time window. Wang et al. then use a dense layer on top of the SDAE to fine-tune this layer with the labeled data to perform the classification. Their method outperforms machine learning algorithms on the Van Kasteren Dataset [43] with the two features representations.

Ghods et al. [44] proposed a method, Activity2Vec to learn an activity Embedding from sensor data. They used a Sequence-to-Sequence model (Seq2Seq) [45] to encode and extract

automatic features from sensors. The model trained as an Autoencoder, to reconstruct the initial input sequence in output. Ghods et al. validate the method with two datasets from HAR domain; one was composed of accelerometer and gyroscope signals from a smartphone, and the other one contained smart sensor events. Their experiment shows that the Activity2Vec method generates good automatic features. They measured the intra-class similarities with handcrafted and Activity2Vec features. It appears that, for the first dataset (smartphone HAR), intra-class similarities are smallest with the Activity2Vec encoding. Conversely, for the second dataset (smart sensors events), the intra-class similarities are smallest with handcrafted features.

### 3.3. Semantics

Previous work has shown that deep learning algorithms, such as Autoencoder or CNN, are capable of extracting features but also of performing classification. Thus, they allow the creation of so-called end-to-end models. However, these models do not translate semantics representing the relationship between activities, as ontologies could represent these relationships [20]. However, in recent years, researchers in the field of Natural Language Processing (NLP) have developed techniques of word embedding and the language model for deep learning algorithms to understand not only the meaning of words but also the structure of phases and texts. A first attempt to add NLP word embedding to deep learning has shown a better performance in daily activity recognition in smart homes [46]. Moreover, the use of the semantics of the HAR domain may allow the development of new learning techniques for quick adaptation, such as zero-shot learning, which is developed in Section 5.

### 3.4. Outlines

All handcrafted methods for extracting features have produced remarkable results in many HAR applications. These approaches assume that each dataset has a set of features that are representative, allowing a learning model to achieve the best performance. However, handcrafted features require extensive pre-processing. This is time consuming and inefficient because the dataset is manually selected and validated by experts. This reduces adaptability to various environments. This is why HAR algorithms must automatically extract the relevant representations.

Methods based on deep learning allow better and higher quality features to be obtained from raw data. Moreover, these features can be learned for any dataset. They can be processed in a supervised or unsupervised manner, for example, windows labeled or not with the name of the activity. In addition, deep learning methods can be end-to-end, i.e., they extract features and perform classification. Thanks to deep learning, great advances have been made in the field of NLP. It allows for representing words, sentences, or texts, thanks to models, structures, and learning methods. These models are able to interpret the semantics of words, to contextualize them, to make prior or posterior correlations between words, and, thus, to increase their performance in terms of sentence or text classification. Moreover, these models are able to automatically extract the right features to accomplish their task. The NLP and HAR domains in smart homes both process data in the form of sequences. In smart homes, sensors generate a stream of events. This stream of events is sequential and ordered, such as words in a text. Some events are correlated to earlier or later events in the stream. This stream of events can be segmented into sequences of activities. These sequences can be similar to sequences of words or sentences. Moreover, semantic links between sensors or types of sensors or activities may exist [20]. We suggest that some of these learning methods or models can be transposed to deal with sequences of sensor events. We think particularly of methods using attention or embedding models.

However, these methods developed for pattern recognition might not be sufficient to analyze these data which are, in fact, temporal series.



## 4. Temporal Data

In a smart home, sensors record the actions and interactions with the residents' environment over time. These recordings are the logs of events that capture the actions and activities of daily life. Most sensors only send their status when there is a change in status, to save battery power and also to not overload wireless communications. In addition, sensors may have different triggering times. This results in scattered sampling of the time series and irregular sampling. Therefore, recognizing human activity in a smart home is a pattern recognition problem in time series with irregular sampling, unlike recognizing human activity in videos or wearables.

In this section, we describe literature methods for segmentation of the sensor data stream in a smart home. These segmentation methods provide a representation of sensor data for human activity recognition algorithms. We highlight the challenges of dealing with the temporal complexity of human activity data in real use cases.

### 4.1. Data Segmentation

As in many fields of activity recognition, a common approach consists of segmenting the data flow. Then, using algorithms to identify the activity in each of these segments. Some methods are more suitable for real-time activity recognition than others. Real time is a necessity to propose reactive systems. In some situations, it is not suitable to recognize activities several minutes or hours after they occur, for example, in case of emergencies, such as fall detection. Quigley et al. [47] have studied and compared different windowing approaches.

#### 4.1.1. Explicit Windowing (EW)

This consists of parsing the data flow per activity [32,33]. Each of these segments corresponds to one window that contain a succession of sensor events belonging to the same activity. This window segmentation depends on the labeling of the data. In the case of absence of labels, it is necessary to find the points of change of activities. The algorithms will then classify these windows by assigning the right activity label. This approach has some drawbacks. First of all, it is necessary to find the segments corresponding to each activity in case of unlabeled data. In addition, the algorithm must use the whole segment to predict the activity. It is, therefore, not possible to use this method in real time.

#### 4.1.2. Time Windows (TW)

The use of TW consists of dividing the data stream into time segments with a regular time interval. This approach is intuitive but rather favorable to the time series of sensors with regular or continuous sampling over time. This is a common technique with wearable sensors, such as accelerometers and gyroscopes. One of the problems is the selection of the optimal duration of the time interval. If the window is too small, it may not contain any relevant information. If it is too large, then the information may be related to several activities, and the dominant activity in the window will have a greater influence on the choice of the label. Van Kasteren et al. [48] determined that a window of 60 s is a time step that allows a good classification rate. This value is used as a reference in many recent works [49–52]. Quigley et al. [47] show that TW achieves a high accuracy but does not allow for finding all classes.

#### 4.1.3. Sensor Event Windows (SEW)

A SEW divides the stream via a sliding window into segments containing an equal number of sensor events. Each window is labeled with the label of the last event in the window. The sensor events that precede the last event in the window define the context of the last event. This method is simple but has some drawbacks. This type of window varies in terms of duration. It is, therefore, impossible to interpret the time between events. However, the relevance of the sensor events in the window can be different, depending on the time interval between the events [53]. Furthermore, because it is a sliding window, it is

possible to find events that belong to the current and previous activity at the same time. In addition, The size of the window in number of events, as for any type of window, is also a difficult parameter to determine. This parameter defines the size of the context of the last event. If the context is too small, there will be a lack of information to characterize the last event. However, if it is too large, it will be difficult to interpret. A window of 20–30 events is usually selected in the literature [34].

#### 4.1.4. Dynamic Windows (DW)

DW uses a non-fixed window size unlike the previous methods. It is a two-stage approach that uses an offline phase and an online phase [54]. In the offline phase, the data stream is split into EW. From the EW, the “best-fit sensor group” is extracted based on rules and thresholds. Then, for the online phase, the dataset is streamed to the classification algorithm. When it identifies the “best-fit sensor group” in the stream, the classifier associates the corresponding label with the given input segment. Problems can arise if the source dataset is not properly annotated. Quigley et al. [47] have shown that this approach is inefficient for modeling complex activities. Furthermore, rules and thresholds are designed by experts, manually, which is time consuming.

#### 4.1.5. Fuzzy Time Windows (FTW)

FTW were introduced in the work of Medina et al. [49]. This type of window was created to encode multi-varied binary sensor sequences, i.e., one series per sensor. The objective is to generate features for each sensor series according to its short, medium, and long term evolution for a given time interval. As for the TW, the FTW segments the signal temporarily. However, unlike other types of window segmentation, FTW use a trapezoidal shape to segment the signal of each sensor. The values defining the trapezoidal shape follow the Fibonacci sequence, which resulted in good performance during classification. The construction of a FTW is done in two steps. First, the sensor stream is resampled by the minute, forming a binary matrix. Each column of this matrix represents a sensor, and each row contains the activation value of the sensor during the minute, i.e., 1 if the sensor is activated in the minute, or 0 otherwise. For each sensor and each minute, a number of FTW is defined and calculated. Thus, each sensor for each minute is represented by a vector translating its activation in the current minute but also its past evolution. The size of this vector is related to the number of FTW. This approach allowed to obtain excellent results for binary sensors. Hamand et al. [50] proposed an extension of FTW by adding FTW using the future data of the sensor in addition to the past information. The purpose of this complement is to introduce a delay in the decision-making of the classifier. The intuition is that relying only on the past is not enough to predict the right label of activity and that, in some cases, delaying the recognition time allows for making a better decision. To illustrate with an example, if a binary sensor deployed on the front door generates an opening activation, the chosen activity could be “the inhabitant has left the house”. However, it may happen that the inhabitant opens the front door only to talk to another person at the entrance of the house and comes back home without leaving. Therefore, the accuracy could be improved by using the activation of the following sensors. It is, therefore, useful to introduce a time delay in decision-making. The longer the delay, the greater the accuracy. However, a problem can appear if this delay is too long, and, indeed, the delay prevents real time. While a long delay may be acceptable for some types of activity, others require a really short decision time in case of an emergency, e.g., the fall of a resident. Furthermore, FTW are only applicable to binary sensors data and do not allow the use of non-binary sensors. However, in a smart home, the sensors are not necessarily binary, e.g., humidity sensors.

#### 4.1.6. Outlines

The table 1 below summarizes and categorizes the different segmentation techniques detailed above.

**Table 1.** Summary and comparison of segmentation methods.

Segmentation Type	Usable for Real Time	Require Resamplig	Time Representation	Usable on Raw Data	Capture Long Term Dependencies	Capture Dependence between Sensors	# Steps
EW	No	No	No	Yes	only inside the sequence	Yes	1
SEW	Yes	No	No	Yes	depends of the size	Yes	1
TW	Yes	Yes	Yes	Yes	depends of the size	No	1
DW	Yes	No	No	Yes	only inside the pre-segmented sequence	Yes	2
FTW	Yes	Yes	Yes	Yes	Yes	No	2

#### 4.2. Time Series Classification

The recognition of human activity in a smart home is a problem of pattern recognition in time series with irregular sampling. Therefore, more specific machine learning for sequential data analysis has also proven efficient for HAR in smart homes.

Indeed, statistical Markov models, such as Hidden Markov Models [29,55] and their generalization, Probabilistic graphical models, such as Dynamic Bayesian Networks [56], can model spatiotemporal information. In the deep learning framework, they have been implemented as Recurrent Neural Networks (RNN). RNN show, today, a stronger capacity to learn features and represent time series or sequential multi-dimensional data.

RNN are designed to take a series of inputs with no predetermined limit on size. RNN remembers the past, and its decisions are influenced by what it has learnt from the past. RNN can take one or more input vectors and produce one or more output vectors, and the output(s) are influenced not just by weights applied on inputs, such as a regular neural network, but also by a hidden state vector representing the context based on prior input(s)/output(s). So, the same input could produce a different output, depending on previous inputs in the series. However, RNN suffers from the long-term dependency problem [57]. To avoid this problem, two RNN variations have been proposed, the Long Short Term Memory (LSTM) [58] and Gated Recurrent Unit (GRU) [59], which is a simplification of the LSTM.

Liciotti et al., in Reference [14], studied different LSTM structures on activity recognition. They showed that the LSTM approach outperforms traditional HAR approaches in terms of classification score without using handcrafted features, as LSTM can generate features that encode the temporal pattern. The higher performance of LSTM was also reported in Reference [60] in comparison traditional machine learning techniques (Naive Bayes, HMM, HSMM, and Conditional Random Fields). Likewise, Sedkly et al. [30] reported that LSTM perform better than AdaBoost, Cortical Learning Algorithm (CLA), Hidden Markov Model, or Multi-layer Perceptron or Structured Perceptron. Nevertheless, the LSTM still have limitations, and their performance is not significantly higher than decision Trees, SVM and stochastic gradient descent of linear SVM, logistic regression, or regression functions. Indeed, LSTM still have difficulties in finding the suitable time scale to balance between long-term temporal dependencies and short term temporal dependencies. A few works have attempted to tackle this issue. Park et al. [61] used a structure using multiple LSTM layers with residual connections and an attention module. Residual connections reduce the gradient vanishing problem, while the attention module marks important events in the time series. To deal with variable time scales, Medina-Quero et al. [49] combined the LSTM with a fuzzy window to process the HAR in real time, as fuzzy windows can automatically adapt the length of its time scale. With accuracies lower than 96%, these refinements still need to be consolidated and improved.

#### 4.3. Complex Human Activity Recognition

Besides, these sequential data analysis algorithms can only process simple, primitive activities, and they cannot yet deal with complex activities. A simple activity is an activity that consists of a single action or movement, such as walking, running, turning on the light, or opening a drawer. A complex activity is an activity that involves a sequence

of actions, potentially involving different interactions with objects, equipment, or other people, for example, cooking.

#### 4.3.1. Sequences of Sub-Activities

Indeed, activities of daily living are not *micro actions*, such as gestures that are carried out the same way by all individuals. Activities of daily living that our smart homes want to recognize can be on the contrary seen as sequences of micro actions, which we can call compound actions. These sequences of micro actions generally follow a certain pattern, but there are no strict constraints on their compositions or the order of micro actions. This idea of compositionality was implemented by an ontology hierarchy of context-aware activities: a tree hierarchy of activities link each activity to its sub-activities [62]. Another work proposed a method to learn this hierarchy: as the Hidden Markov Model approach is not well suited to process long sequences, an extension of HMM called Hierarchical Hidden Markov Model was proposed in Reference [63] to encode multi-level dependencies in terms of time and follow a hierarchical structure in their context. To our knowledge, there have been no extensions of such hierarchical systems using deep learning, but hierarchical LSTM using two-layers of LSTM to tackle the varying composition of actions for HAR based on videos proposing [64] or using two hidden layers in the LSTM for HAR using wearables [65] can constitute inspirations for HAR in smart home applications. Other works in video-based HAR proposed to automatically learn a stochastic grammar describing the hierarchical structure of complex activities from annotations acquired from multiple annotators [66].

The idea of these HAR algorithms is to use the context of a sensor activation, either by introducing multi-timescale representation to take into account longer term dependencies or by introducing context-sensitive information to channel the attention in the stream of sensor activations.

The latter idea can be developed much further by taking advantage of the methods developed by the field of natural language processing, where texts also have a multi-level hierarchical structure, where the order of words can vary and where the context of a word is very important. Embedding techniques, such as ELMo [67], based on LSTM, or, more recently, BERT [68], based on Transformers [69], have been developed to handle sequential data while handling long-range dependencies through context-sensitive embeddings. These methods model the context of words to help the processing of long sequences. Applied to HAR, they could model the context of the sensors and their order of appearance. Taking inspiration from Reference [46,66], we can draw a parallel between NLP and HAR: a word is apparent to a sensor event, a micro activity composed of sensor events is apparent to a sentence, and a compound activity composed of sub-activities is a paragraph. The parallel between word and sensor events has led to the combination of word encodings with deep learning to improve the performance of HAR in smart homes in Reference [46].

#### 4.3.2. Interleave and Concurrent Activities

Human activities are often carried out in a complex manner. Activities can be carried out in an interleave or concurrent manner. An individual may alternately cook and wash dishes, or cook and listen to music simultaneously, but could just as easily cook and wash dishes, alternately, while listening to music. The possibilities are infinite in terms of activity scheduling. However, some activities seem impossible to see appearing in the dataset and could be anomalous, such as cooking while the individual sleeps in his room.

Researchers are working on this issue. Modeling this type of activity is becoming complex. However, it could be modeled as a multi-label classification problem. Safyan et al. [70] have explored this problem using ontology. Their approach uses a semantic segmentation of sensors and activities. This allows the model to relate the possibility that certain activities may or may not occur at the same time for the same resident. Li et al. [71] exploit a CNN-LSTM structure to recognize concurrent activity with multi-modal sensors.

### 4.3.3. Multi-User Activities

Moreover, monitoring the activities of daily living performed by a single resident is already a complex task. The complexity increases with several residents. The same activities become more difficult to recognize. On the one hand, in a group, a resident may interact to perform common activities. In this case, the activation of the sensors reflects the same activity for each resident in the group. On the other hand, everyone can perform different activities simultaneously. This produces a simultaneous activation of the sensors for different activities. These activations are then merged and mixed in the activity sequences. An activity performed by one resident is a noise for the activities of another resident.

Some researchers are interested in this problem. As with the problem of recognizing competing activities, the multi-resident activity recognition problem is a multi-label classification problem [72]. Tran et al. [73] tackled the problem using a multi-label RNN. Natani et al. [74] studied different neural network architectures, such as MLP, CNN, LSTM, GRU, or hybrid structures, to evaluate which structure is the most efficient. The hybrid structure that combines a CNN 1D and a LSTM is the best performing one.

### 4.4. Outlines

A number of algorithms have been studied for HAR in smart homes. Table 2 show a summary and comparison of recent HAR methods in smart homes.

**Table 2.** Summary and comparison of activity recognition methods in smart homes.

Ref	Segmentation	Data Representation	Encoding	Feature Type	Classifier	Dataset	Real-Time
[14]	EW	Sequence	Integer sequence (one integer for each possible sensors activations)	Automatic	Uni LSTM, Bi LSTM, Cascade LSTM, Ensemble LSTM, Cascade Ensemble LSTM	CASAS [75]: Milan, Cairo, Kyoto2, Kyoto3, Kyoto4	No
[60]	TW	Multi-channel	Binary matrix	Automatic	Uni LSTM	Kasteren [43]	Yes
[61]	EW	Sequence	Integer sequence (one integer for each sensor Id)	Automatic	Residual LSTM, Residual GRU	MIT [76]	No
[49]	FTW	Multi-channel	Real values matrix (computed values inside each FTW)	Manual	LSTM	Ordonez [77], CASAS A & CASAS B [75]	Yes
[15]	EW + SEW	Multi-channel	Binary picture	Automatic	2D CNN	CASAS [75]: Aruba	No
[51]	FTW	Multi-channel	Real values matrix (computed values inside each FTW)	Manual	Joint LSTM + 1D CNN	Ordonez [77], Kasteren [43]	Yes
[41]	TW	Multi-channel	Binary matrix	Automatic	1D CNN	Kasteren [43]	Yes
[78]	TW	Multi-channel/Sequence	Binary matrix, Binary vector, Numerical vector, Probability vector	Automatic/Manual Automatic/Manual	Autoencoder, 1D CNN, 2D CNN, LSTM, DBN	Ordonez [77]	Yes
[34]	SEW	Sequence	Categorical values	Manual	Random Forest	CASAS [75]: HH101-HH125	Yes

LSTM shows excellent performance on the classification of irregular time series in the context of a single resident and simple activities. However, human activity is more complex than this. In addition, challenges related to the recognition of concurrent, interleaved or idle activities offer more difficulties. Previous cited works did not take into account these type of activities. Moreover, people rarely live alone in a house. This is why even more complex challenges are introduced, including the recognition of activity in homes with multiple residents. These challenges are multi-class classification problems and still unsolved.

In order to address these challenges, activity recognition algorithms should be able to segment the stream for each resident. Techniques in the field of image processing based on Fully Convolutional Networks [79], such as U-Net [80], allow for segmentation of the images. These same approaches can be adapted to time series [81] and can constitute inspirations for HAR in smart home applications.

## 5. Data Variability

Not only are real human activities complex, the application of human activity recognition in smart homes for real-use cases also faces issues causing a discrepancy between training and test data. The next subsections detail the issues inherent to smart homes: the temporal drift of the data and the variability of settings.

### 5.1. Temporal Drift

Smart homes through their sensors and interactions with residents collect data on the behavior of residents. Initial training data is the portrait of the activities performed at the time of registration. A model is generated and trained using this data. Over time, the behavior and habits of the residents may change. The data that is now captured is no longer the same as the training data. It corresponds to a time drift as introduced in Reference [82]. This concept means that the statistical properties of the target variable, which the model is trying to predict, evolve over time in an unexpected way. A shift in the distribution between the training data and the test data.

To accommodate this drift, algorithms for HAR in smart homes should incorporate life-long learning to continuously learn and adapt to changes in human activities from new data as proposed in Reference [83]. Recent works in life-long learning incorporating deep learning as reviewed in Reference [84] could help tackle this issue of temporal drift. In particular, one can imagine that an interactive system can from time to time request labeled data to users to continue to learn and adapt. Such algorithms have been developed under the names of interactive reinforcement learning or active imitation learning in robotics. In Reference [85], they allowed the system to learn micro and compound actions, while minimizing the number of requests for labeled data by choosing when, what information to ask, and to whom to ask for help. Such principles could inspire a smart home system to continue to adapt its model, while minimizing user intervention and optimizing his intervention, by pointing out the missing key information.

### 5.2. Variability of Settings

Besides these long-term evolutions, the data from one house to another are also very different, and the model learned in one house is hardly applicable in another because of the change in house configuration, sensors equipment, and families' compositions and habits. Indeed, the location, the number and the sensors type of smart homes can influence activity recognition systems performances. Each smart homes can be equipped in different ways and have different architecture in terms of sensors, room configuration, appliance, etc. Some can have a lot of sensors, multiple bathrooms, or bedrooms and contain multiple appliances, while others can be smaller, such as a single apartment, where sensors can be fewer and have more overlaps and noisy sequences. Due to this difference in house configurations, a model that optimized in the first smart homes could perform poorly in another. This issue could be solved by collecting a new dataset for each new household to train the models anew; however, this is costly, as explained in Section 6.

Another solution is to adapt the models learned in a household to another. Transfer learning methods have recently been developed to allow pre-trained deep learning models to be used with different data distributions, as reviewed in Reference [86]. Transfer learning using deep learning has been successfully applied to time series classification, as reviewed in Reference [87]. For activity recognition, Cook et al. [88] reviewed the different types of knowledge that could be transferred in traditional machine learning. These methods can be updated with deep learning algorithms and by benefiting from recent advances in transfer learning for deep learning. Furthermore, adaptation to new settings have recently been improved by the development of meta-learning algorithms. Their goal is to train a model on a variety of learning tasks, so it can solve new learning tasks using only a small number of training samples. This field has seen recent breakthroughs, as reviewed in Reference [89], which has never been applied yet to HAR. Yet, the peculiar variability of data of HAR in smart homes can only benefit from such algorithms.

## 6. Datasets

Datasets are key to train, test, and validate activity recognition systems. Datasets were first generated in laboratories. However, these records do not allow enough variety and complexity of activities and were not real enough. To overcome these issues, public datasets were created from recordings in real homes with volunteer residents. In parallel

to being able to compare in the same condition and on the same data, some competitions were created, such as Evaluating AAL Systems Through Competitive Benchmarking-AR (EvAAL-AR) [90] or UCAMl Cup [91].

However, the production of datasets is a tedious task and recording campaigns are difficult to manage. They require volunteer actors and apartments or houses equipped with sensors. In addition, data annotation and post-processing take a lot of time. Intelligent home simulators have been developed as a solution to generate datasets.

This section presents and analyzes some real and synthetic datasets in order to understand the advantages and disadvantages of these two approaches.

### 6.1. Real Smart Home Dataset

A variety of public real homes datasets exist [43,75,76,92,93]. De-la-Hoz et al. [94] provides an overview of sensor-based datasets used in HAR for smart homes. They compiled documentation and analysis of a wide range of datasets with a list of results and applied algorithms. However, such dataset production implies some problems as: sensors type and placement, variability in terms of user profile or typology of dwelling, and the annotation strategy.

#### 6.1.1. Sensor Type and Positioning Problem

When acquiring data in a house, it is difficult to choose the sensors and their numbers and locations. It is important to select sensors that are as minimally invasive as possible in order to respect the privacy of the volunteers [92]. No cameras nor video recordings were used. The majority of sensor-oriented smart home datasets use so-called low-level sensors. These include infrared motion sensors (PIR), magnetic sensors for openings and closures, pressure sensors placed in sofas or beds, sensors for temperature, brightness, monitoring of electricity or water consumption, etc.

The location of these sensors is critical to properly capture activity. Strategic positioning allows for accurate capture of certain activities, e.g., a water level sensor in the toilet to capture toilet usage or a pressure sensor under a mattress to know if a person is in bed. There is no precise method or strategy for positioning and installing sensors in homes. CASAS [75] researchers have proposed and recommended a number of strategic positions. However, some of these strategic placements can be problematic in terms of evolution. It is possible to imagine that, during the life of a house, the organization or use of its rooms changes, e.g., if a motion sensor is placed above the bed to capture its use. However, if the bed is moved to a different place in the room, then the sensor will no longer be able to capture this information. In the context of a dataset and the use of the dataset to validate the algorithms, this constraint is not important. However, it becomes important in the context of real applications to evaluate the resilience of algorithms, which must continue to function in case of loss of information.

In addition to positioning, it is important to choose enough sensors to cover a maximum of possible activities. The number of sensors can be very different from one dataset to another. For example, the MIT dataset [76] uses 77 and 84 sensors for each of these apartments. The Kasteren dataset [43] uses between 14 and 21 sensors. ARAS [92] has apartments with 20 sensors. Orange4Home [93] is based on an apartment equipped with 236 sensors. This difference can be explained by the different types of dwellings but also by the number and granularity of the activities that we want to recognize. Moreover, some dataset are voluntarily over-equipped. There is still no method nor strategy to define the number of sensors installed according to an activity list.

#### 6.1.2. Profile and Typology Problem

It is important to take into account that there are different typologies of houses: apartment, house, with garden, with floors, without floor, one or more bathrooms, one or more bedrooms, etc. These different types and variabilities of houses lead to difficulties, such as: the possibility that the same activity takes place in different rooms, that the

investment in terms of number of sensors can be more or less important, or that the network coverage of the sensors can be problematic. For example, Alerndar et al. [92] faced a problem of data synchronization. One of their houses required two sensor networks to cover the whole house. They must synchronize the data for dataset needs. It is, therefore, necessary that the datasets can propose different house configurations in order to evaluate the algorithms in multiple configurations. Several datasets with several houses exist [43, 43,76,92]. CASAS [75] is one of them, with about 30 several houses configurations. These datasets are very often used in the literature [94]. However, the volunteers are mainly elderly people, and coverage of several age groups is important. A young resident does not have the same behavior as an older one. The Orange4Home dataset [93] covers the activity of a young resident. The number of residents is also important. The activity recognition is more complex in the case of multiple residents. This is why several datasets also cover this field of research [43,75,92].

### 6.1.3. Annotation Problem

Dataset annotation is something essential for supervised algorithm training. When creating, these datasets, it is necessary to deploy strategies to enable this annotation, such as journal [43], smartphone applications [93], personal digital assistant (PDA) [76], Graphical User Interface (GUI) [92], or voice records, to annotate the dataset [43].

As these recordings are made directly by volunteers, they are asked to annotate their own activities. For the MIT dataset [76], residents used a PDA to annotate their activities. Every 15 min, the PDA beeped to prompt residents to answer a series of questions to annotate their activities; however, several problems were encountered with this method of user self-annotation. However, several problems were encountered with this method of self-annotation by the user, such as some short activities not being entered, errors in label selection, or omissions. A post-annotation based on the study of a posteriori activations was necessary to overcome these problems, thus potentially introducing new errors. In addition, this annotation strategy is cumbersome and stressful because of the frequency of inquiries. It requires great rigor from the volunteer and, at the same time, interrupts activity execution by pausing it when the information is given. These interruptions reduce the fluidity and natural flow of activities.

Van Kasteren et al. [43] proposed another way of annotating their data. The annotation was also done by the volunteers themselves, although using voice through a Bluetooth headset and a journal. This strategy allowed the volunteers to be free to move around and not need to create breaks in the activities. This allowed for more fluid and natural sequences of activities. The Diary allowed the volunteers to complete some additional information when wearing a helmet was not possible. However, wearing a helmet all day long remains a constraint.

The volunteers of the ARAS dataset [92] used a simple Graphical User Interface (GUI) to annotate their activities. Several instances were placed in homes to minimize interruptions in activities and avoid wearing an object, such as a helmet, all day long. Volunteers were asked to indicate only the beginning of each activity. It is assumed that residents will perform the same activity until the next start of the activity. This assumption reflects a bias that sees human activity as a continuous stream of known activity.

## 6.2. Synthetic Smart Home Dataset

The cost to build real smart homes and the collection of datasets for such scenarios is expensive and sometimes infeasible for many projects. Measurements campaigns should include a wide variety of activities and actors. It should be done with sufficient rigor to obtain qualitative data. Moreover, finding the optimal placement of the sensors [95], finding appropriate participants [96,97], and the lack of flexibility [98,99] makes the dataset collection difficult. For these reasons, researchers imagined smart homes simulation tools [100].



These simulation tools can be categorized into two main approaches, model-based [101] and interactive [102], according to Synnott et al. [103]. The model-based approach uses predefined models of activities to generate synthetic data. In contrast, the interactive approach relies on having an avatar that can be controlled by a researcher, human participant, or simulated participant. Some hybrid simulators, such as OpenSH [100], can combine advantages from both interactive and model-based approaches. In addition, a smart homes simulation tool can focus on the dataset generation or data visualization. Some simulation tools provide multi-resident or fast forwarding to accelerate the time during execution.

These tools allow you to quickly generate data and visualize it. However, the capture of activities can be unnatural and not noisy. Some uncertainty may be missing.

### 6.3. Outlines

All these public datasets, synthetic or real, are useful and allow evaluating processes. Both, show advantages and drawbacks. Table 3 details some datasets from the literature, resulting from the hard work of the community.

**Table 3.** Example of real datasets of the literature.

Ref	Multi-Resident	Resident Type	Duration	Sensor Type	# of Sensors	# of Activity	# of Houses	Year
[43]	No	Elderly	12–22 days	Binary	14–21	8	3	2011
[92]	Yes	Young	2 months	Binary	20	27	3	2013
[93]	No	Young	2 weeks	Binary, Scalar	236	20	1	2017
[75]	Yes	Elderly	2–8 months	Binary, Scalar	14–30	10–15	>30	2012
[77]	No	Elderly	14–21 days	Binary	12	11	2	2013
[76]	No	Elderly	2 weeks	Binary, Scalar	77–84	9–13	2	2004

Real datasets, such as Orange4Home [93], provide a large sensor set. That can help to determine which sensors can be useful for which activity. CASAS [75] proposes many houses or apartment configurations and topologies with elderly people, which allows evaluating the adaptability to house topologies. ARAS [92] proposes younger people and multi-residents’ living environments, which is useful to validate the noisy resilience and segmentation ability of the activity recognition system. The strength of real datasets is their variability, as well as their representativeness in number and execution of activities. However, sensors can be placed too strategically and wisely chosen to cover some specific kinds of activities. In some datasets, PIR sensors are used as a grid or installed as a checkpoint to track residents trajectory. Strategic placement, a large number of sensors, or the choice of a particular sensor is great to help algorithms to infer knowledge, but they are not the real ground truth.

Synthetic datasets allow for quick evaluation of different configuration sensors and topologies. In addition, they can produce large amounts of data without real setup or volunteer subjects. The annotation is more precise compared to real dataset methods (diary, smartphone apps, voice records).

However, activities provided by synthetic datasets are less realistic in terms of execution rhythm and variability. Every individual has its own rhythm in terms of action duration, interval or order. The design of the virtual smart homes can be a tedious task for a non-expert designer. Moreover, no synthetic datasets are publicly available. Only some dataset generation tools, such as OpenSH [100], are available.

Today, even if smart sensors become cheaper and cheaper, real houses are not equipped with a wide range of sensors as can be found in datasets. It is not realistic to find an opening sensor on a kitchen cabinet. Real homes contains PIR to monitor wide areas with the security system. Temperature sensors to control the heat. More and more air qualitative or luminosity sensors can be found. Some houses are now equipped with smart lights or smart plugs. Magnetic sensors can be found on external openings. In addition, now,

some houses provide general electrical and water consumption. These datasets are not representative of the actual home sensor equipment.

Another issue, as shown above, is the annotation. Supervised algorithms need qualitative labels to learn correct features and classify activities. Residents' self-annotation can produce errors and lack of precision. Post-processing to add annotations adds uncertainty, as they are always based on hypothesis, such as every activity being performed sequentially. However, the human activity flow is not always sequential. Very few datasets provide concurrent or interleaved activities. Moreover, every dataset proposes its own taxonomy for annotations, even if synthetic datasets try to overcome annotation issues.

This section demonstrates the difficulty of providing a correct evaluation system or dataset. In addition, the work already provided by all the scientific community is excellent. Thanks to this amount of work, it is possible to, in certain conditions, evaluate activity recognition systems.

However, there are several areas of research that can be explored to help the field progress more quickly. A first possible research axis for data generation is the generation of data from video games. Video games constitute a multi-billion dollar industry, where developers put great effort into building highly realistic worlds. Recent works in the field of semantic video segmentation consider and use video games to generate datasets in order to train algorithms [104,105]. Recently, Roitberg et al. [106] studied a first possibility using a commercial game by Electronic Arts (EA), "The Sims 4", a daily life simulator game, to reproduce the video Toyota SmartHome dataset [107]. The objective was to evaluate and train HAR algorithms from video produced by a video game and compare them to the original dataset. This work showed promising results. An extension of this work could be envisaged in order to generate datasets of sensor activity traces. Moreover, every dataset proposes its own taxonomy. Some are inspired by medical works, such as, Katz et al.'s work [108], to define a list of basic and necessary activities. However, there is no proposal for a hierarchical taxonomy, e.g., cook lunch and cook dinner are children activities of cook, or taxonomy taking into account concurrent or parallel activities. The suggestion of a common taxonomy for datasets is a research axis to be studied in order to homogenize and compare algorithms more efficiently.

## 7. Evaluation Methods

In order to validate the performance of the algorithms, the researchers use datasets. However, learning the parameters of a prediction function and testing it on the same data is a methodological error: a model that simply repeats the labels of the samples it has just seen would have a perfect score but could not predict anything useful on data that is still invisible. This situation is called overfitting. To avoid it, it is common practice in a supervised machine learning experiment to retain some of the available data as a dataset for testing. Several methods exist in the field of machine learning and deep learning. For the problem of HAR in smart houses, some of them have been used by researchers.

The evaluation of these algorithms is not only related to the use of these methods. It depends on the methodology but also on the datasets on which the evaluation is based. It is not uncommon that pre-processing is necessary. However, this pre-processing can influence the final results. This section highlights some of the biases that can be induced by pre-processing the datasets, as well as the application and choice of certain evaluation methods.

### 7.1. Datasets Pre-Processing

#### 7.1.1. Unbalanced Datasets Problem

Unbalanced datasets pose a challenge because most of the machine learning algorithms used for classification have been designed assuming an equal number of examples for each class. This results in models with poor predictive performance, especially for the minority class. This is a problem because, in general, the minority class is larger; therefore, the problem is more sensitive to classification errors for the minority class than for the majority class. To get around this problem, some researchers will rebalance the dataset by removing

classes that are too little represented and by randomly removing examples for the most represented classes [15]. These approaches allow for increasing the performance of the algorithms but do not allow them to represent the reality.

Within the context of the activities of daily life, certain activities are performed more or less often during the course of the days. A more realistic approach is to group activities under a new, more general label; for example, “preparing breakfast”, “preparing lunch”, “preparing dinner”, and “preparing a snack” can be grouped under the label “preparing a meal”. Therefore, activities that are less represented but semantically close can be used as parts of example. This can allow fairer comparisons between datasets if the label names are shared. Liciotti et al. [14] adopted this approach to compare several datasets between them. One of the drawbacks is the loss of granularity of activities.

#### 7.1.2. The Other Class Issue

In the field of HAR in smart houses, it is very frequent that a part of the dataset is not labeled. Usually, the label “Other” is assigned to these unlabeled events. The class “Other” generally represents 50% of the dataset [14,16]. This makes it the most represented class in the dataset and unbalances the dataset. Furthermore, the “Other” class may represent several different activity classes or simply something meaningless. Some researchers choose to suppress this class, as it is judged to be over-represented and containing too many random sequences. Others prefer to remove it from the training phase and, therefore, from the training set. However, they keep it in the test set in order to evaluate the system in a more real-life environment [33]. Yala et al. [33] evaluated performance with and without the “Other” class and showed that this choice has a strong impact on the final results.

However, being able to dissociate this class opens perspectives. Algorithms able to isolate these sequences could propose to the user to annotate them in the future in order to discover new activities.

#### 7.1.3. Labeling Issue

As noted above, the datasets for the actual houses are labeled by the residents themselves, via a logbook or graphical user interface. They are then post-processed by the responsible researchers. However, it is not impossible that some labels may be missing, as in the CASAS Milan dataset [75]. Table 4 presents an extract from the Milan dataset where labels are missing. However, events or days are duplicated, i.e., same timestamp, same sensor, same value, and same activity label. A cleaning of the dataset must be considered before the algorithms are formed. Obviously, depending on the quality of the labels and data, the results will be different. Indeed, some occurrence of classes could be artificially increased or decreased. Some events could be labeled “Other”, even though they actually belong to a defined activity. In this case, the recognition algorithm could label this event correctly, but it would appear to be confused with another class in the confusion matrix.

**Table 4.** CASAS [75] Milan dataset anomaly.

Date	Time	Sensor ID	Value	Label
2010-01-05	08:25:37.000026	M003	OFF	
2010-01-05	08:25:45.000001	M004	ON	Read begin
...	...	...	...	...
2010-01-05	08:35:09.000069	M004	ON	
2010-01-05	08:35:12.000054	M027	ON	
2010-01-05	08:35:13.000032	M004	OFF	(Read should end)
2010-01-05	08:35:18.000020	M027	OFF	
2010-01-05	08:35:18.000064	M027	ON	
2010-01-05	08:35:24.000088	M003	ON	
2010-01-05	08:35:26.000002	M012	ON	(Kitchen Activity should begin)
2010-01-05	08:35:27.000020	M023	ON	
...	...	...	...	...
2010-01-05	08:45:22.000014	M015	OFF	
2010-01-05	08:45:24.000037	M012	ON	Kitchen Activity end
2010-01-05	08:45:26.000056	M023	OFF	

#### 7.1.4. Evaluation Metrics

Since HAR is a multi-class classification problem, researchers use metrics [109], such as Accuracy, Precision, Recall, and F-Score, to evaluate their algorithms [41,49,61]. These metrics are defined by means of four features, such as true Positive, true Negative, false Positive, and false Negative, of class  $C_i$ . The F-score, also called the F1-score, is a measure of a model’s accuracy on a dataset. The F-score is a way of combining the Precision and Recall of the model, and it is defined as the harmonic mean of the model’s Precision and Recall. It should not be forgotten that real house datasets are mostly imbalanced in terms of class. In other words, some activities have more examples than others and are in the minority. In an imbalanced dataset, a minority class is harder to predict because there are few examples of this class, by definition. This means it is more challenging for a model to learn the characteristics of examples from this class, as well as to differentiate examples from this class from the majority class. Therefore, it would be more appropriate to use metrics weighted by the class support of the dataset, such as balanced Accuracy, weighted Precision, weighted Recall, or weighted F-score [110,111].

#### 7.2. Evaluation Process

##### 7.2.1. Train/Test

A first way to evaluate the algorithms is to divide the datasets into two distinct parts: one for training and the other for testing. It is generally chosen to use 70% for training and 30% for testing. Several researchers have chosen to adopt this method. Surong et al. [16] adopted this evaluation method in the application of real time activation recognition. In order to show the generalization of their approach, they chose to divide the datasets temporally into two equal parts. Then, they chose to re-divide each of these parts temporally into training and test datasets. Thus, they propose two sub-sets of training and test. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. Moreover, it does not allow for taking into account the drift [34] of the activities. In addition, it is always possible that the algorithm overfitted on the test sets because the parameters were adapted to optimal values. This approach does not guarantee a generalization of the algorithms.

##### 7.2.2. K-Fold Cross Validation

This is a wide approach used for model evaluation. It consists of dividing the dataset into K sub-dataset; the value of K is often between 3 and 10. K-1 dataset is selected for training, and the remaining dataset for testing. The algorithm iterates until all the sub-dataset is used for testing. The average of the training K scores is used to evaluate the generalization of the algorithm. It is usually customary that the data is mixed before

being divided into K sub-datasets in order to increase the generalization capability of the algorithms. However, it is possible that some classes are not represented in the training or test sets. That is why some implementations propose that all classes are represented in tests as not training.

In the context of HAR in smart homes, this method is a good approach for classification of EW [14,61]. Indeed, EW can be considered as independent and not temporally correlated. However, it seems not relevant for sliding windows, especially if they have a strong overlap, and the windows are distributed equally according to their class between the test and training set. The training and test sets would look too similar, which would increase the performance of the algorithms and would not allow it to generalize enough.

### 7.2.3. Leave-One-Out Cross-Validation

This is a special case of cross-validation where the number of folds equals the number of instances in the dataset. Thus, the learning algorithm is applied once for each instance, using all other instances as a training set and using the selected instance as a single-item test set.

Singh et al. [60] and Medina-Quero et al. [49] used this validation method in a context of real-time HAR. In their experiments, the dataset is divided into days. One day is used for testing, while the other days are used for training. Each day becomes a test day, in turn. This approach allows a large part of the dataset to be used for training, as well as allowing the algorithms to train on a wide variety of data. However, the size of the test is not very significant and does not allow for demonstrating the generalization of the algorithm in the case of HAR in smart homes.

### 7.2.4. Multi-Day Segment

Aminikhanghahi et al. [34] propose a validation method called Multi-Day Segment. This approach proposes to take into account the sequential nature of segmentation in a context of real-time HAR. Indeed, in this real-time context, each segment or window is temporally correlated. According to Aminikhanghahi et al., and as expressed above, cross-validation would bias the results in this context. A possible solution would be to use the 2/3 training and 1/3 test partitioning, as described above. However, this introduces the concept of drift into the data. Drift in terms of change in resident behavior would induce a big difference between the training and test set.

To overcome these problems, the proposed method consists of dividing the dataset into 6 consecutive days. The first 4 days are used for training, and the last 2 days are used for testing. This division into 6-day segments creates a rotation that allows for representing every day of the week in the training and test set. In order to make several folds, the beginning of the 6-day sequence is shifted 1 day forward at each fold. This approach allows for maintaining the order of the data, while avoiding the drift of the dataset.

## 7.3. Outlines

Different validation methods for HAR in smart homes were reviewed in this section, as shown in Table 5. Depending on the problem being addressed, not all methods can be used to evaluate an algorithm.

**Table 5.** Summary of methods for evaluating activity recognition algorithms.

Ref	Train/Test Spilt	K-Fold Cross-Validation	Leave-One-Out Cross-Validation	Multi-Day Segment	Respect Time Order of Activities	Sensitive to Data Drift Problem	Real Time Recognition	Offline Recognition	Usable on Small Datasets
[16]	✓				Yes	Yes	Yes	Yes	No
[14,15,61]		✓			No	No	No	Yes	No
[41,49,51,60,78]			✓		Not necessarily	No	Yes	Yes	Yes
[34]				✓	Yes	No	Yes	Yes	No

In the case of offline HAR, i.e., with EW or pre-segmented activity sequences, the K-fold cross-validation seems to be the most suitable, provided that the time dependency between segments is not taken into account. Otherwise, it is preferable to use another method. The Leave-One-Out Cross-Validation approach is an alternative. It allows for processing of datasets containing little data. However, the days are considered as independent. It is not possible to make a link between two different days, e.g., a weekday or a weekend day. Aminikhanghahi et al. [34] proposed a method to preserve the temporal dependence of the segments and avoid the problem of data drift induced by changes in the habits of the resident(s) over time.

In addition, the pre-processing of dataset data, the rebalancing, the removal of the “Other” class, and the annotation of events affect the algorithms’ performance. It is, therefore, important to take into account the evaluation method and the pre-processing performed, in order to judge the performance of the algorithm. Moreover, classic metrics, such as accuracy or F score, may not be sufficient. It may be more judicious to use metrics weighted by the number of representations if dataset classes, such as dataset, are unbalanced. Balanced accuracy, or F1 weighted score, should be a better metric in this case [110,111].

A major problem in the area of HAR in smart homes is the lack of evaluation protocols. Establishing a uniform protocol according to the type of problem to be solved (real-time, offline) would speed up research in this field and allow a fairer comparison between the proposed algorithms and approaches.

## 8. General Conclusion and Discussions

In this article, we have highlighted the challenges of Human Activity Recognition in smart homes, some of which have particularities compared to other fields of HAR. We have proposed a taxonomy of the main components of a human activity recognition algorithm and reviewed the most promising solutions. To overcome the current issues, we point out the opportunities provided by new advances from other fields.

### 8.1. Comparison with Other HAR

While human activity recognition algorithms have seen tremendous improvements for vision-based data owing to the rapid development of deep learning for image processing, human activity recognition using wearables and sensors on objects are also seeing significant improvements. However, vision-based systems are seen by users as too intrusive as these systems could unveil too much private information, whereas wearables and sensors on objects require the daily instrumentation of the sensors on the body of subjects or their personal objects, and ambient sensors could provide a solution to tackle this issue.

HAR in smart homes have seen recent advances owing to the development of recent deep learning algorithms for end-to-end classification, such as convolutional neural networks. It also benefits from recent algorithms for sequence learning, such as long-short term memory, but, as with video processing, sequence learning still needs to be improved to both be able to deal with the vanishing gradient problem and to take into account the context of the sensor readings. The temporal dimension is incidentally a particularity of ambient sensor systems, as the data for a sparse and irregular time series. The irregular sampling in time has also been tackled with adapted windowing methods for data segmentation. In addition to the time windows used in other HAR fields, sensor event windows are also commonly used. The sparsity of the data of ambient sensors does not allow machine learning algorithms to take advantage of the redundancy of data over time, as in the case of videos where successive video frames are mostly similar. Moreover, whereas HAR in videos, the context of the human action can be seen in the images by the detection of his environment or objects of attention, the sparsity of the HAR in ambient sensors results in a high reliance in the past information to infer the context information.

While HAR in ambient sensors have to face the problems of complex activities, such as sequences of activities, concurrent activities, or multi-occupant activities, or data drift,

it also has to tackle specific unsolved problems, such as the variability of data. Indeed, the data collected by sensors are even more sensitive to the house configuration, the choice of sensors, and their localization.

### 8.2. Taxonomy and Challenges

To face its specific challenges and the challenges common to other systems, in our review, we introduced a taxonomy of the main components of a human activity recognition algorithm for real-use. The three components we have pointed out are: classification, automatic feature extraction, and time series analysis. It needs to carry out a pattern recognition from raw data, thus requiring feature extraction. Moreover, the algorithm must integrate a time series analysis.

While pattern recognition analysis and the feature extraction challenges seem to be well tackled by deep learning algorithms, such as CNN, the sequence analysis parts have improved recently with the application of LSTM. Both approaches based on CNN and LSTM are reported to give equivalent performance levels, and state-of-the-art developments are mostly based on either LSTM or convolutional deep learning. However, the sequence analysis challenges still remain largely unsolved because of the impact of the sparsity and irregularity of the data on context understanding and long-term reasoning. In particular, it makes the challenges of composite activities (sequences of activities), concurrent activities, multi-user activities recognition, and data drift more difficult. The sparsity of the data also makes it more difficult to cope with the variability of the smart home data in its various settings.

According to our analysis, the state-of-the-art options in HAR for ambient sensors are still far from ready to be deployed in real-use cases. To achieve this, the field must address the shortcomings of datasets, as well as needs to also standardize the evaluation metrics so as to reflect the requirements for a real-use deployment and to enable fair comparison between algorithms.

### 8.3. Opportunities

Moreover, we believe that recent advances in machine learning from other fields also offer opportunities for significant advances in HAR in smart homes.

We advocate that the application of recent NLP techniques can bring advances in solving some of these challenges. Indeed, NLP also deploys methods of sequence analysis and has also seen tremendous advances in the recent years. For instance, sparsity of the data can be alleviated by a better domain knowledge in the form of an emerging semantic. Thus, taking inspiration from word encoding and language models, we can automatically introduce semantic knowledge between activities, as shown in the preliminary study of Reference [46]. Furthermore, a semantic encoding of the data will also help the system be more robust to unknown data as in the challenges of data drift or adaptation to changes, as it could be able to relate new data semantically to known data. Besides, the recent techniques for analyzing long texts by inferring long-term context, but also analyzing the sequences of words and sentences, can serve as an inspiration to analyze sequences of activities or composite activities.

Lastly, we think that the unsolved problem of adaptation to changes of habits, users, or sensor sets could soon find its solution in the current research on meta learning and interactive learning.

### 8.4. Discussion

In this review, we have pointed out the key elements for an efficient algorithm of human activity recognition in smart homes. We have also pointed out the most efficient methods, in addition to the remaining challenges and present opportunities. However, the full deployment of smart home services, beyond the HAR algorithms, depends also on the development of the hardware systems and the acceptability and usability of these systems by final users.

For the hardware systems, the development of IoT devices with the improvement in the accuracy and autonomy, along with the decrease in their cost, will make them accessible

to normal households. Despite cheaper sensors and actuators, it will not be realistic to provide all homes with a large set of sensors as in the current datasets, but real homes are not as lavishly equipped. Thus, a smart home system needs to optimize their hardware under constraints of budget, house configuration, number of inhabitants, etc. Smart home builder companies need to provide an adequate HAR hardware kit. To determine the minimal set of sensors, recently, Bolleddula et al. [112] used PCA to determine the most important sensors in a lavishly equipped smart home. This study is a first work to imagine a minimal sensors setup.

Finally, while IoT devices seem to be better accepted by users than cameras, there are still social barriers to the adoption of smart homes that need to be overcome [113]. These require a trustworthy privacy-preserving data management, as well as reliable cyber-secure systems.

**Author Contributions:** Damien Bouchabou performed the literature reading, algorithm coding and measurements. Sao Mai Nguyen provided guidance, structuring and proofreading of the article. Christophe Lohr, Ioannis Kanellos and Benoit Leduc provided guidance and proofreading. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** The work is partially supported by project VITAAL and is financed by Brest Metropole, the region of Brittany and the European Regional Development Fund (ERDF). This work was carried out within the context of a CIFRE agreement with the company Delta Dore in Bonemain 35270 France, managed by the National Association of Technical Research (ANRT) in France.

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

## References

1. Chan, M.; Estève, D.; Escriba, C.; Campo, E. A review of smart homes—Present state and future challenges. *Computer methods and programs in biomedicine* **2008**, *91*, 55–81.
2. Hussain, Z.; Sheng, M.; Zhang, W.E. Different Approaches for Human Activity Recognition: A Survey. *arXiv preprint arXiv:1906.05074* **2019**.
3. Dang, L.M.; Min, K.; Wang, H.; Piran, M.J.; Lee, C.H.; Moon, H. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition* **2020**, *108*, 107561.
4. Beddiar, D.R.; Nini, B.; Sabokrou, M.; Hadid, A. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications* **2020**, *79*, 30509–30555.
5. Singh, D.; Psychoula, I.; Kropf, J.; Hanke, S.; Holzinger, A. Users' perceptions and attitudes towards smart home technologies. *International Conference on Smart Homes and Health Telematics*. Springer, 2018, pp. 203–214.
6. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115.
7. Li, X.; Zhang, Y.; Marsic, I.; Sarcevic, A.; Burd, R.S. Deep learning for rfid-based activity recognition. *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, 2016, pp. 164–175.
8. Gomes, L.; Sousa, F.; Vale, Z. An intelligent smart plug with shared knowledge capabilities. *Sensors* **2018**, *18*, 3961.
9. Chen, L.; Hoey, J.; Nugent, C.D.; Cook, D.J.; Yu, Z. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **2012**, *42*, 790–808.
10. Aggarwal, J.; Xi, L. Human activity recognition from 3d data: A review. *Pattern Recognition Letters* **2014**, *48*, 70–80.
11. Vrigkas, M.; Nikou, C.; Kakadiaris, I.A. A review of human activity recognition methods. *Frontiers in Robotics and AI* **2015**, *2*, 28.
12. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* **2019**, *119*, 3–11.
13. Chen, K.; Zhang, D.; Yao, L.; Guo, B.; Yu, Z.; Liu, Y. Deep learning for sensor-based human activity recognition: overview, challenges and opportunities. *arXiv preprint arXiv:2001.07416* **2020**.
14. Liciotti, D.; Bernardini, M.; Romeo, L.; Frontoni, E. A Sequential Deep Learning Application for Recognising Human Activities in Smart Homes. *Neurocomputing* **2019**. doi:10.1016/j.neucom.2018.10.104.
15. Gochoo, M.; Tan, T.H.; Liu, S.H.; Jean, F.R.; Alnajjar, F.S.; Huang, S.C. Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and DCNN. *IEEE journal of biomedical and health informatics* **2018**, *23*, 693–702.
16. Yan, S.; Lin, K.J.; Zheng, X.; Zhang, W. Using latent knowledge to improve real-time activity recognition for smart IoT. *IEEE Transactions on Knowledge and Data Engineering* **2019**.
17. Perkowski, M.; Philipose, M.; Fishkin, K.; Patterson, D.J. Mining models of human activities from the web. *Proceedings of the 13th international conference on World Wide Web*, 2004, pp. 573–582.



18. Chen, L.; Nugent, C.D.; Mulvenna, M.; Finlay, D.; Hong, X.; Poland, M. A logical framework for behaviour reasoning and assistance in a smart home. *International Journal of Assistive Robotics and Mechatronics* **2008**, *9*, 20–34.
19. Chen, L.; Nugent, C.D. *Human Activity Recognition and Behaviour Analysis*; Springer, 2019.
20. Yamada, N.; Sakamoto, K.; Kunito, G.; Isoda, Y.; Yamazaki, K.; Tanaka, S. Applying ontology and probabilistic model to human activity recognition from surrounding things. *IPSJ Digital Courier* **2007**, *3*, 506–517.
21. Chen, L.; Nugent, C.; Mulvenna, M.; Finlay, D.; Hong, X. Semantic smart homes: towards knowledge rich assisted living environments. In *Intelligent Patient Management*; Springer, 2009; pp. 279–296.
22. Chen, L.; Nugent, C. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems* **2009**.
23. Chen, L.; Nugent, C.D.; Wang, H. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering* **2011**, *24*, 961–974.
24. Logan, B.; Healey, J.; Philipose, M.; Tapia, E.M.; Intille, S. A long-term evaluation of sensing modalities for activity recognition. International conference on Ubiquitous computing. Springer, 2007, pp. 483–500.
25. Vail, D.L.; Veloso, M.M.; Lafferty, J.D. Conditional random fields for activity recognition. Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, 2007, pp. 1–8.
26. Fleury, A.; Vacher, M.; Noury, N. SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. *IEEE transactions on information technology in biomedicine* **2009**, *14*, 274–283.
27. Brdiczka, O.; Crowley, J.L.; Reignier, P. Learning situation models in a smart home. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2008**, *39*, 56–63.
28. van Kasteren, T.; Krose, B. Bayesian activity recognition in residence for elders. 2007 3rd IET International Conference on Intelligent Environments. IET, 2007, pp. 209–212.
29. Cook, D.J. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems* **2010**, *2010*, 1.
30. SEDKY, M.; HOWARD, C.; Alshammari, T.; Alshammari, N. Evaluating machine learning techniques for activity classification in smart home environments. *International Journal of Information Systems and Computer Sciences* **2018**, *12*, 48–54.
31. Chinellato, E.; Hogg, D.C.; Cohn, A.G. Feature space analysis for human activity recognition in smart environments. 2016 12th International Conference on Intelligent Environments (IE). IEEE, 2016, pp. 194–197.
32. Cook, D.J.; Krishnan, N.C.; Rashidi, P. Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics* **2013**, *43*, 820–828.
33. Yala, N.; Fergani, B.; Fleury, A. Feature extraction for human activity recognition on streaming data. 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA). IEEE, 2015, pp. 1–6.
34. Aminikhanghahi, S.; Cook, D.J. Enhancing activity recognition using CPD-based activity segmentation. *Pervasive and Mobile Computing* **2019**, *53*, 75–89.
35. Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M.P.; Shyu, M.L.; Chen, S.C.; Iyengar, S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)* **2018**, *51*, 1–36.
36. Fang, H.; He, L.; Si, H.; Liu, P.; Xie, X. Human activity recognition based on feature selection in smart home using back-propagation algorithm. *ISA transactions* **2014**, *53*, 1629–1638.
37. Irvine, N.; Nugent, C.; Zhang, S.; Wang, H.; Ng, W.W. Neural network ensembles for sensor-based human activity recognition within smart environments. *Sensors* **2020**, *20*, 216.
38. Tan, T.H.; Gochoo, M.; Huang, S.C.; Liu, Y.H.; Liu, S.H.; Huang, Y.F. Multi-resident activity recognition in a smart home using RGB activity image and DCNN. *IEEE Sensors Journal* **2018**, *18*, 9718–9727.
39. Mohmed, G.; Lotfi, A.; Pourabdollah, A. Employing a deep convolutional neural network for human activity recognition based on binary ambient sensor data. Proceedings of the 13th ACM International Conference on Pervasive Technologies Related to Assistive Environments, 2020, pp. 1–7.
40. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, pp. 1097–1105.
41. Singh, D.; Merdivan, E.; Hanke, S.; Kropf, J.; Geist, M.; Holzinger, A. Convolutional and recurrent neural networks for activity recognition in smart environment. In *Towards integrative machine learning and knowledge extraction*; Springer, 2017; pp. 194–205.
42. Wang, A.; Chen, G.; Shang, C.; Zhang, M.; Liu, L. Human activity recognition in a smart home environment with stacked denoising autoencoders. International conference on web-age information management. Springer, 2016, pp. 29–40.
43. van Kasteren, T.L.; Englebienne, G.; Kröse, B.J. Human activity recognition from wireless sensor network data: Benchmark and software. In *Activity recognition in pervasive intelligent environments*; Springer, 2011; pp. 165–186.
44. Ghods, A.; Cook, D.J. Activity2vec: Learning adl embeddings from sensor data with a sequence-to-sequence model. *arXiv preprint arXiv:1907.05597* **2019**.
45. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 2014, pp. 3104–3112.
46. Bouchabou, D.; Nguyen, S.M.; Lohr, C.; Kanellos, I.; Leduc, B. Fully Convolutional Network Bootstrapped by Word Encoding and Embedding for Activity Recognition in Smart Homes. IJCAI 2020 Workshop on Deep Learning for Human Activity Recognition, 2021.

47. Quigley, B.; Donnelly, M.; Moore, G.; Galway, L. A Comparative Analysis of Windowing Approaches in Dense Sensing Environments. *Proceedings* **2018**, *2*, 1245. doi:10.3390/proceedings2191245.
48. van Kasteren, T.L.M.; others. Activity recognition for health monitoring elderly using temporal probabilistic models. *ASCI*, 2011.
49. 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 Systems with Applications* **2018**, *114*, 441–453.
50. Hamad, R.A.; Hidalgo, A.S.; Bouguelia, M.R.; Estevez, M.E.; Quero, J.M. Efficient activity recognition in smart homes using delayed fuzzy temporal windows on binary sensors. *IEEE journal of biomedical and health informatics* **2019**, *24*, 387–395.
51. Hamad, R.A.; Yang, L.; Woo, W.L.; Wei, B. Joint learning of temporal models to handle imbalanced data for human activity recognition. *Applied Sciences* **2020**, *10*, 5293.
52. Hamad, R.A.; Kimura, M.; Yang, L.; Woo, W.L.; Wei, B. Dilated causal convolution with multi-head self attention for sensor human activity recognition. *Neural Computing and Applications* **2021**, pp. 1–18.
53. Krishnan, N.C.; Cook, D.J. Activity recognition on streaming sensor data. *Pervasive and mobile computing* **2014**, *10*, 138–154.
54. Al Machot, F.; Mayr, H.C.; Ranasinghe, S. A windowing approach for activity recognition in sensor data streams. 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN). *IEEE*, 2016, pp. 951–953.
55. Cook, D.J.; Schmitter-Edgecombe, M. Assessing the quality of activities in a smart environment. *Methods of information in medicine* **2009**, *48*, 480.
56. Philipose, M.; Fishkin, K.P.; Perkowitz, M.; Patterson, D.J.; Fox, D.; Kautz, H.; Hahnel, D. Inferring activities from interactions with objects. *IEEE pervasive computing* **2004**, *3*, 50–57.
57. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* **1994**, *5*, 157–166.
58. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.
59. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* **2014**.
60. Singh, D.; Merdivan, E.; Psychoula, I.; Kropf, J.; Hanke, S.; Geist, M.; Holzinger, A. Human activity recognition using recurrent neural networks. International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer, 2017, pp. 267–274.
61. Park, J.; Jang, K.; Yang, S.B. Deep neural networks for activity recognition with multi-sensor data in a smart home. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). *IEEE*, 2018, pp. 155–160.
62. Hong, X.; Nugent, C.; Mulvenna, M.; McClean, S.; Scotney, B.; Devlin, S. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing* **2009**, *5*, 236 – 252. doi:<https://doi.org/10.1016/j.pmcj.2008.05.002>.
63. Asghari, P.; Soelimani, E.; Nazerfard, E. Online Human Activity Recognition Employing Hierarchical Hidden Markov Models, 2019, [[arXiv:cs.LG/1903.04820](https://arxiv.org/abs/cs.LG/1903.04820)].
64. Devanne, M.; Papadakis, P.; Nguyen, S.M. Recognition of Activities of Daily Living via Hierarchical Long-Short Term Memory Networks. International Conference on Systems Man and Cybernetics. *IEEE*, 2019, pp. 3318–3324. doi:10.1109/SMC.2019.8914457.
65. Wang, L.; Liu, R. Human Activity Recognition Based on Wearable Sensor Using Hierarchical Deep LSTM Networks. *Circuits, Systems, and Signal Processing* **2020**, *39*, 837–856. doi:10.1007/s00034-019-01116-y.
66. Tayyub, J.; Hawasly, M.; Hogg, D.C.; Cohn, A.G. Learning Hierarchical Models of Complex Daily Activities from Annotated Videos. *IEEE Winter Conf. on Applications of Computer Vision*, 2018, pp. 1633–1641.
67. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* **2018**.
68. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**.
69. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv preprint arXiv:1706.03762* **2017**.
70. Safyan, M.; Qayyum, Z.U.; Sarwar, S.; García-Castro, R.; Ahmed, M. Ontology-driven semantic unified modelling for concurrent activity recognition (OSCAR). *Multimedia Tools and Applications* **2019**, *78*, 2073–2104.
71. Li, X.; Zhang, Y.; Zhang, J.; Chen, S.; Marsic, I.; Farneth, R.A.; Burd, R.S. Concurrent activity recognition with multimodal CNN-LSTM structure. *arXiv preprint arXiv:1702.01638* **2017**.
72. Alhamoud, A.; Muradi, V.; Böhnstedt, D.; Steinmetz, R. Activity recognition in multi-user environments using techniques of multi-label classification. *Proceedings of the 6th International Conference on the Internet of Things*, 2016, pp. 15–23.
73. Tran, S.N.; Zhang, Q.; Smallbon, V.; Karunanithi, M. Multi-resident activity monitoring in smart homes: A case study. 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). *IEEE*, 2018, pp. 698–703.
74. Natani, A.; Sharma, A.; Perumal, T. Sequential neural networks for multi-resident activity recognition in ambient sensing smart homes. *Applied Intelligence* **2021**, pp. 1–15.
75. Cook, D.J.; Crandall, A.S.; Thomas, B.L.; Krishnan, N.C. CASAS: A smart home in a box. *Computer* **2012**, *46*, 62–69.

76. Tapia, E.M.; Intille, S.S.; Larson, K. Activity recognition in the home using simple and ubiquitous sensors. *International conference on pervasive computing*. Springer, 2004, pp. 158–175.
77. Ordóñez, F.; De Toledo, P.; Sanchis, A.; others. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors* **2013**, *13*, 5460–5477.
78. Wang, A.; Zhao, S.; Zheng, C.; Yang, J.; Chen, G.; Chang, C.Y. Activities of Daily Living Recognition With Binary Environment Sensors Using Deep Learning: A Comparative Study. *IEEE Sensors Journal* **2020**, *21*, 5423–5433.
79. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
80. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
81. Perslev, M.; Jensen, M.H.; Darkner, S.; Jennum, P.J.; Igel, C. U-time: A fully convolutional network for time series segmentation applied to sleep staging. *arXiv preprint arXiv:1910.11162* **2019**.
82. Schlimmer, J.C.; Granger, R.H. Incremental learning from noisy data. *Machine learning* **1986**, *1*, 317–354.
83. Thrun, S.; Pratt, L. *Learning to Learn*; Springer US: Boston, MA, 1998.
84. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks* **2019**, *113*, 54–71.
85. Duminy, N.; Nguyen, S.M.; Zhu, J.; Duhaut, D.; Kerdreux, J. Intrinsically Motivated Open-Ended Multi-Task Learning Using Transfer Learning to Discover Task Hierarchy. *Applied Sciences* **2021**, *11*. doi:10.3390/app11030975.
86. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *Journal of Big Data* **2016**, *3*, 9. doi:10.1186/s40537-016-0043-6.
87. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Transfer learning for time series classification. 2018 IEEE international conference on big data (Big Data). IEEE, 2018, pp. 1367–1376.
88. Cook, D.; Feuz, K.; Krishnan, N. Transfer Learning for Activity Recognition: A Survey. *Knowledge and information systems* **2013**, *36*, 537–556. doi:10.1007/s10115-013-0665-3.
89. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-Learning in Neural Networks: A Survey, 2020.
90. Gjoreski, H.; Kozina, S.; Gams, M.; Lustrek, M.; Álvarez-García, J.A.; Hong, J.H.; Ramos, J.; Dey, A.K.; Bocca, M.; Patwari, N. Competitive live evaluations of activity-recognition systems. *IEEE Pervasive Computing* **2015**, *14*, 70–77.
91. Espinilla, M.; Medina, J.; Nugent, C. UCAM Cup. Analyzing the UJA Human Activity Recognition Dataset of Activities of Daily Living. *Proceedings* **2018**, *2*, 1267. doi:10.3390/proceedings2191267.
92. Alemdar, H.; Ertan, H.; Incel, O.D.; Ersoy, C. ARAS human activity datasets in multiple homes with multiple residents. 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops. IEEE, 2013, pp. 232–235.
93. Cumin, J.; Lefebvre, G.; Ramparany, F.; Crowley, J.L. A dataset of routine daily activities in an instrumented home. *International Conference on Ubiquitous Computing and Ambient Intelligence*. Springer, 2017, pp. 413–425.
94. De-La-Hoz-Franco, E.; Ariza-Colpas, P.; Quero, J.M.; Espinilla, M. Sensor-based datasets for human activity recognition—a systematic review of literature. *IEEE Access* **2018**, *6*, 59192–59210.
95. Helal, S.; Kim, E.; Hossain, S. Scalable approaches to activity recognition research. *Proceedings of the 8th international conference pervasive workshop*, 2010, pp. 450–453.
96. Helal, S.; Lee, J.W.; Hossain, S.; Kim, E.; Hagrais, H.; Cook, D. Persim-Simulator for human activities in pervasive spaces. 2011 Seventh International Conference on Intelligent Environments. IEEE, 2011, pp. 192–199.
97. Mendez-Vazquez, A.; Helal, A.; Cook, D. Simulating events to generate synthetic data for pervasive spaces. *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*. Citeseer, 2009.
98. Armac, I.; Retkowitz, D. Simulation of smart environments. *IEEE International Conference on Pervasive Services*. IEEE, 2007, pp. 257–266.
99. Fu, Q.; Li, P.; Chen, C.; Qi, L.; Lu, Y.; Yu, C. A configurable context-aware simulator for smart home systems. 2011 6th International Conference on Pervasive Computing and Applications. IEEE, 2011, pp. 39–44.
100. Alshammari, N.; Alshammari, T.; Sedky, M.; Champion, J.; Bauer, C. Openshs: Open smart home simulator. *Sensors* **2017**, *17*, 1003.
101. Lee, J.W.; Cho, S.; Liu, S.; Cho, K.; Helal, S. Persim 3d: Context-driven simulation and modeling of human activities in smart spaces. *IEEE Transactions on Automation Science and Engineering* **2015**, *12*, 1243–1256.
102. Synnott, J.; Chen, L.; Nugent, C.D.; Moore, G. The creation of simulated activity datasets using a graphical intelligent environment simulation tool. 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2014, pp. 4143–4146.
103. Synnott, J.; Nugent, C.; Jeffers, P. Simulation of smart home activity datasets. *Sensors* **2015**, *15*, 14162–14179.
104. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for data: Ground truth from computer games. *European conference on computer vision*. Springer, 2016, pp. 102–118.
105. Richter, S.R.; Hayder, Z.; Koltun, V. Playing for benchmarks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2213–2222.
106. Roitberg, A.; Schneider, D.; Djamal, A.; Seibold, C.; Reiß, S.; Stiefelhagen, R. Let’s Play for Action: Recognizing Activities of Daily Living by Learning from Life Simulation Video Games. *arXiv preprint arXiv:2107.05617* **2021**.

107. Das, S.; Dai, R.; Koperski, M.; Minciullo, L.; Garattoni, L.; Bremond, F.; Francesca, G. Toyota smarthome: Real-world activities of daily living. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 833–842.
108. Katz, S. Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living. *Journal of the American Geriatrics Society* **1983**, *31*, 721–727.
109. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information processing & management* **2009**, *45*, 427–437.
110. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. *Learning from imbalanced data sets*; Vol. 10, Springer, 2018.
111. He, H.; Ma, Y. *Imbalanced learning: foundations, algorithms, and applications* **2013**.
112. Bolleddula, N.; Hung, G.Y.C.; Ma, D.; Noorian, H.; Woodbridge, D.M.k. Sensor Selection for Activity Classification at Smart Home Environments. 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2020, pp. 3927–3930.
113. Balta-Ozkan, N.; Davidson, R.; Bicket, M.; Whitmarsh, L. Social barriers to the adoption of smart homes. *Energy Policy* **2013**, *63*, 363–374.