Hindawi
Mobile Information Systems
Volume 2021, Article ID 6961343, 11 pages
https://doi.org/10.1155/2021/6961343

Research Article

A Deep Learning-Based Framework for Human Activity Recognition in Smart Homes

Alaeddine Mihoub

Department of Management Information Systems and Production Management, College of Business and Economics, Qassim University, P.O. Box: 6640, Buraidah 51452, Saudi Arabia

Correspondence should be addressed to Alaeddine Mihoub; a.mihoub@qu.edu.sa

Received 18 May 2021; Revised 4 August 2021; Accepted 20 August 2021; Published 11 September 2021

Academic Editor: Anand Nayyar

Human behavior modeling in smart environments is a growing research area treating several challenges related to ubiquitous computing, pattern recognition, and ambient assisted living. Thanks to recent progress in sensing devices, it is now possible to design computational models able of accurate detection of residents' activities and daily routines. For this goal, we introduce in this paper a deep learning-based framework for activity recognition in smart homes. This framework proposes a detailed methodology for data preprocessing, feature mining, and deep learning techniques application. The novel framework was designed to ensure a deep exploration of the feature space since three main approaches are tested, namely, the all-features approach, the selection approach, and the reduction approach. Besides, the framework proposes the evaluation and the comparison of several well-chosen deep learning techniques such as autoencoder, recurrent neural networks (RNN), and some of their derivatives models. Concretely, the framework was applied on the "Orange4Home" dataset which represents a recent dataset specially designed for smart homes research. Our main findings show that the best approach for efficient classification is the selection approach. Furthermore, our overall results outperformed baseline models based on random forest classifiers and the principal component analysis technique, especially the results of our RNN-based model for the all-features approach and the results of our autoencoder-based model for the feature reduction approach.

1. Introduction

Analyzing human routines in ambient and smart environments represents a growing research area due to its multiple scientific, engineering, and data-privacy challenges [1]. Smart homes are a typical example of these intelligent environments. They represent classic living houses equipped with multiple sensors measuring various modalities such as energy consumption, luminosity, lights status, doors opening, human movements, etc. [2]. These sensing technologies have opened new opportunities to understand humans' behavior and model their routines. In the healthcare field, daily routines, describing daily human activities, are known as Activities of Daily Living (ADLs) [3]. For instance, cleaning, sleeping, eating, using stairs, cooking, and showering are some typical examples of these ADLs. Thanks to smart homes, it is henceforth possible to monitor resident's ADLs, track their vital status, evaluate life quality, and enhance their well-being [4]. Indeed, one goal of these augmented environments is to assist elderly and disabled people to have decent and autonomous living [5, 6]. More generally, smart homes sensing technologies come with several benefits not only for persons with special needs. They help in optimizing energy consumption, guarantee house security, improve general comfort, and enhance entertainment applications within the house [7].

Our main challenge in this research is to build computational models to efficiently recognize ADLs in smart home livings. The long-term goal of this work is to build user-aware houses able to analyze humans' behavior and respond to their daily needs. To reach this aim, our methodology in this research is to mine the complete process of ADLs detection task: from data preprocessing to the evaluation of many machine learning approaches. To ensure this deep mining of the detection task, we propose in this paper an original framework based on several data mining techniques and deep learning approaches. The suggested deep learning-based framework should enable smart processing of multimodal data, a deep analysis of human behavior subtleties, and an accurate detection of activity patterns. Compared to other works, it can be considered as a powerful guideline for researchers and engineers to process any ADL classification task, especially on similar datasets format.

The rest of the paper is organized as follows: the next section reviews the state of the art of ADLs recognition approaches. In Section 3, we propose a rapid introduction to the main deep learning approaches applied in this work. In Section 4, we introduce our proposed framework and its different compartments. In Section 5, we detail our experiments: we present the used dataset and how the proposed approaches in the framework were applied and implemented. All recognition results and main findings are discussed in Section 6. Section 7 summarizes our contributions and concludes the paper.

2. Related Work

Activity recognition is a foundation stone of a myriad of real-world applications such as social robots [8, 9], personnel skills analytics [10–12], sports analytics [13, 14], education analytics [15], group interaction analysis [16, 17], affective computing [18], human behavior understanding [19, 20], assisted living and healthcare [5], and security and well-being in smart homes [7]. For these applications, ADLs recognition is considered as a classification problem [4]. It consists of inferring resident's activities based on the smart home sensors or cameras. In our research, we focus particularly on the sensors-based detection approach [21]. Two types of sensors are widely used: (1) wearable sensors such as smart watches and helmets [22] and (2) ambient sensors that are installed directly within the house or its compartments [23, 24]. Both sensors share the same objective of catching all possible human actions and interactions. Data generated from these sensors may suffer from several defects such as noise, redundancies, missing values, and other imperfections. Before any classification step, a preprocessing step is then crucial to deal with these problems [25] but also to unify data types and rescale numeral values. Researchers should also decide which features to use by either selecting a subset of features, extracting new features [26], or reducing the feature space [27]. Afterward comes the classification task, which consists of simply mapping the input features to the right output activity. At this step, several metrics should be used to control and evaluate the accuracy of this operation.

Initially, ADLs classification literature considered traditional approaches of machine learning [28] such as support vector machines (SVM) [5, 29], Naïve Bayes [30], random forests (RF) [23, 31], Ensemble approaches [32], and hidden Markov models (HMM) [33–35] (see Table 1). However, in recent years, a major shift is observed since the rapid development of deep learning techniques [42]. In fact, conventional techniques require heavy handcrafted feature exploration based on researcher knowledge [43], which limits model development and extensions. Besides, they are inefficient in capturing highly complex activities composed of micromovements sequences and gestures [44]. On the opposite side, deep learning techniques can automatically infer and extract relevant features reducing then complex handcrafted operations. Moreover, recent results of these methods have shown unparalleled performance in many related fields to sequence classification [45] such as object recognition, speech recognition, natural language processing, and CyberSecurity [42, 46]. For all these reasons, deep learning has received a lot of attention in many recent works [36–38, 47–49] and represents, with no doubt, a promising approach for ADLs recognition and classification.

Table 1: Human activity recognition models.

Traditional approaches	Deep learning approaches
Support vector machines [5, 29]	Deep belief network [36]
Naïve Bayes [30]	RNN [37]
Random forests [23, 31]	CNN, LSTM [38, 39]
Ensemble approaches [32]	CNN-LSTM [40]
Hidden Markov models [33–35]	CNN–Bi-LSTM [41]

For instance, using available smart home datasets for activity recognition, researchers in [38] compared deep learning techniques such as convolutional neural networks (CNN) and Long Short-Term Memory (LSTM), to other traditional machine learning approaches. Experimental results show similar results from CNN and LSTM but both classifiers outperform other classic models. In [36], the authors addressed the classification problem by applying the Deep Belief Network (DBN) model on data recorded from resident's wearable devices. The paper shows the effectiveness of the proposed approach compared to other classic approaches such as SVM (support vector machines). In [37], activity recognition for elderly people with dementia was explored using many types of recurrent neural networks (RNN). Obtained results showed promising and competitive performance compared to classic state-of-the-art techniques. Other works demonstrating the efficiency of deep learning approaches for ADLs detection task can be found in [39–41, 47–49].

Consequently, we have chosen in our proposed framework to apply multiple deep learning approaches well-known for their efficiency, such as MLP (Multilayer Perceptron), deep autoencoder, RNN (Recurrent Neural Networks), LSTM (Long Short-Term Memory), and GRU (Gated Recurrent Units). To compare these techniques in different configurations related to feature extraction and reduction, our proposed framework was carefully designed to ensure an exhaustive evaluation of all possible approaches. Indeed, the proposed deep learning framework englobes and mines the whole process of the activity detection task from data preprocessing to the application of deep learning models and their final evaluation. This framework was applied to an interesting dataset in the field of smart homes called the "Orange4Home dataset" [24]. The advantages of this recent dataset can be found in Experiments section. One challenge of our framework is to outperform the classification rates of baseline models computed on the same dataset. Actually, our main contributions can be listed as follows:

(i) The introduction of a deep learning-based framework for activity recognition: this framework proposes a detailed methodology for data preprocessing, feature selection, and feature reduction as well as the training/testing steps for the chosen deep learning models. Note that this framework can be easily generalized and applied to other similar datasets' format.

(ii) The enhancement of recognition rates computed with baseline models based on random forest classifiers and the principal component analysis technique.

(iii) The testing of multiple approaches to ensure a deep exploration of the feature space. These approaches are the all-features approach, the feature selection approach, and the feature reduction approach.

(iv) The evaluation and comparison of several well-chosen deep learning techniques (MLP, autoencoder, RNN, LSTM, and GRU). From our findings, several conclusions were drawn concerning the best approaches for relevant ADLs detection.

In the next section, we start by briefly presenting the main concepts behind neural networks and deep learning.

3. Deep Learning Models

3.1. ANN. Deep Learning [42] is a branch of machine learning that relies on artificial neural network architectures (ANNs) [50]. ANNs are computational models that are inspired by the biological neurons of the human brain attempting to simulate similar information processing and task performing. ANNs are composed of multiple interconnected and successive layers. Each layer is represented by a set of artificial neurons called nodes. Each node is connected to the next layer nodes via links, the same way as natural neurons and their connections. Each link is represented by a numeric weight corresponding to the impact of each node on the next layer node. Nodes are typically organized in multiple successive layers. The first layer is called the input layer since it receives input data that will be injected into the network. The intermediate layers are known as the hidden layers and the last one is called the output layer. The output layer typology depends essentially on the network task whether it is regression or classification, etc. The basic form of ANNs is also known as Multilayer Perceptron (MLP) or Feedforward Network (see Figure 1).

Like the rest of machine learning models, MLP models can be learned using data samples. The training process in MLP consists of adjusting connections weights to better handle the desired task. In supervised learning, this training process is ensured by the backpropagation algorithm [50]. Technically it consists of minimizing a loss function that computes observed errors from the output layer. Then, with respect to the loss function, network weights are updated. MLP has shown to be a good technique in many classification or regression problems [51]; however, it may not be the most appropriate approach to handle sequential and temporal input data.

3.2. RNN, LSTM, and GRU. For time-series data, the most appropriate models are recurrent neural networks (RNN) [52] and their derivatives since they were particularly developed to model temporal and sequential data. In its basic form, RNN is a basic Feedforward Network in which a hidden layer receives, in addition to actual inputs, the outputs of the hidden layer calculated in the previous time step (see Figure 1). The recurrent hidden layer as shown in Figure 1 represents a temporal layer that allows the information to flow from one step to another. This information is known as the hidden state of the RNN. That is why it is often said that RNN, thanks to their hidden state, have a certain kind of memory that remembers what has been computed. However, when RNN processes long sequences, it shows more difficulties in retaining information from old steps. This problem is known as the vanishing gradient problem [53, 54]. The vanishing gradient problem causes the RNN to be a short-term memory network unable to learn long-term dependencies. To overcome these issues, two models were developed which are LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Units) [52].

LSTM and GRU are derived from RNN but present new internal mechanisms and gates concepts allowing for regulating the flow of information over time. These concepts allow retaining previous information as in RNN but with more efficiency in carrying relevant information from earlier time steps. For LSTM models, the core principles rely on carrying two pieces of information through the processing which are a cell state and a hidden state. Actually, an LSTM cell contains three gates: a forget gate to decide which information should be deleted or retained, an input gate to update the cell state, and an output gate to decide the next hidden state. For GRU, the process is simplified by letting down the cell state and retaining only a hidden state over processing. Moreover, it has only two gates: a reset gate and an update gate. The update gate decides how to combine the previous memory with actual input while the reset gate determines how much of the previous memory should be kept. More details on RNN, LSTM, and GRU can be found on [52].

3.3. Autoencoders. Another type of ANN is autoencoders [55]. An autoencoder (denoted as AE) is considered as a Feedforward Network similar to MLP since it contains at least three layers: one input layer, one or more hidden layers, and an output layer. "Deep autoencoder" notation is used when more than one hidden layer is used. The major difference with MLP is that the latter model tries to predict a target variable, while AE aims to reconstruct the input data with a minimum of errors. To achieve this goal, the input and the output layers must be identical in the learning phase. Moreover, an AE can be considered as a dimension reduction technique trying to extract an optimal compressed representation of the initial data. Concretely, an AE chains

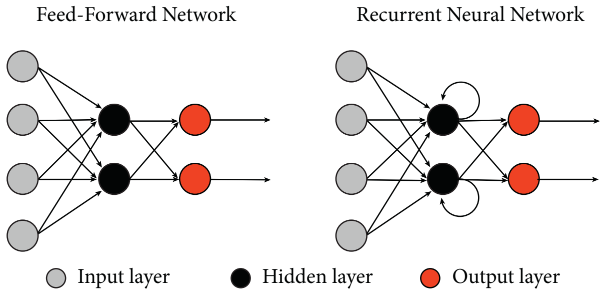

Figure 1: (a) Topologies of Feedforward Network and (b) the recurrent neural network.

two main stages: the encoding stage and the decoding stage (see an example in Figure 2). In the encoding phase, the network reduces the number of nodes from the input layer to the middle hidden layer, called the bottleneck layer. This bottleneck layer represents the new compressed representation of the input data. In the decoding phase, the network increases the number of nodes from the bottleneck to the output layer to retrieve the initial input data. Encoding and decoding mechanisms have made AE approach a great data-driven technique for learning and extracting relevant features avoiding many troubles related to handcrafted features [56]. Furthermore, since neural networks apply nonlinear transformation between layers, AE has shown great performance for feature reduction compared to other state-of-the-art reduction techniques [47]. In the next section, we present our proposed framework in which we detail the different proposed approaches.

4. Proposed Framework

The proposed deep learning framework (see Figure 3) explores the whole process of the activity detection task. It was carefully designed to ensure an exhaustive evaluation of all possible approaches related to feature choice and the application of deep learning models. In fact, it proposes detailed guidelines for the following:

(i) Data preprocessing
(ii) Features to use
(iii) Models building

Each of those steps is detailed in the following subsections. Note that this framework can be generalized and applied to any smart home dataset built over ambient sensors and devices.

4.1. Data Preprocessing. From a smart home setting, each sensor data could be stored in one file. Each file contains two columns: timestamp and sensor value. Time granularity is different from one sensor to another. Our preprocessing approach aims at merging all data sensors to one matrix in which row index is timestamp and column indexes are sensor labels. The output matrix should not contain any missing values and all values must be normalized. To reach this goal, we propose a specific preprocessing algorithm that chains essentially three big steps (see Figure 3):

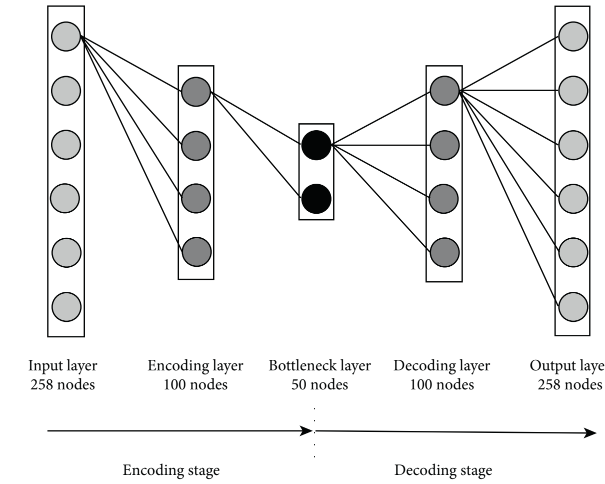

Figure 2: The autoencoder model applied to Orange4Home for feature reduction purposes.

(i) Data fusion: this first manipulation is effectuated by the fusion of all timestamps from all sensors. Some missing values may appear since sensors do not share the exact timestamps.

(ii) Filling missing values: many strategies are proposed to replace missing values with relevant ones. First, missing values should be replaced by the last valid observation from the past. Next, the remaining ones, especially in the beginning, are replaced by zero in case of numerical sensor and by the first valid observation in case of a categorical sensor. This way all missing values are suitably treated.

(iii) Data normalization: for many machine learning models, it is recommended and even required in some cases to normalize data to a standard format. For this goal, a min-max scaler is firstly applied to numerical values. Secondly, Binarization is applied for two-category variables (e.g., “ON” and “OFF” are, respectively, replaced by 1 and 0). Lastly, hot encoding is applied for multiple-categories variables (i.e., three categories or higher).

Therefore, applying these three steps, we should come out with a homogenous numerical table in which all columns’ values are ranged between 0 and 1. These table columns represent the initial input features of the next steps.

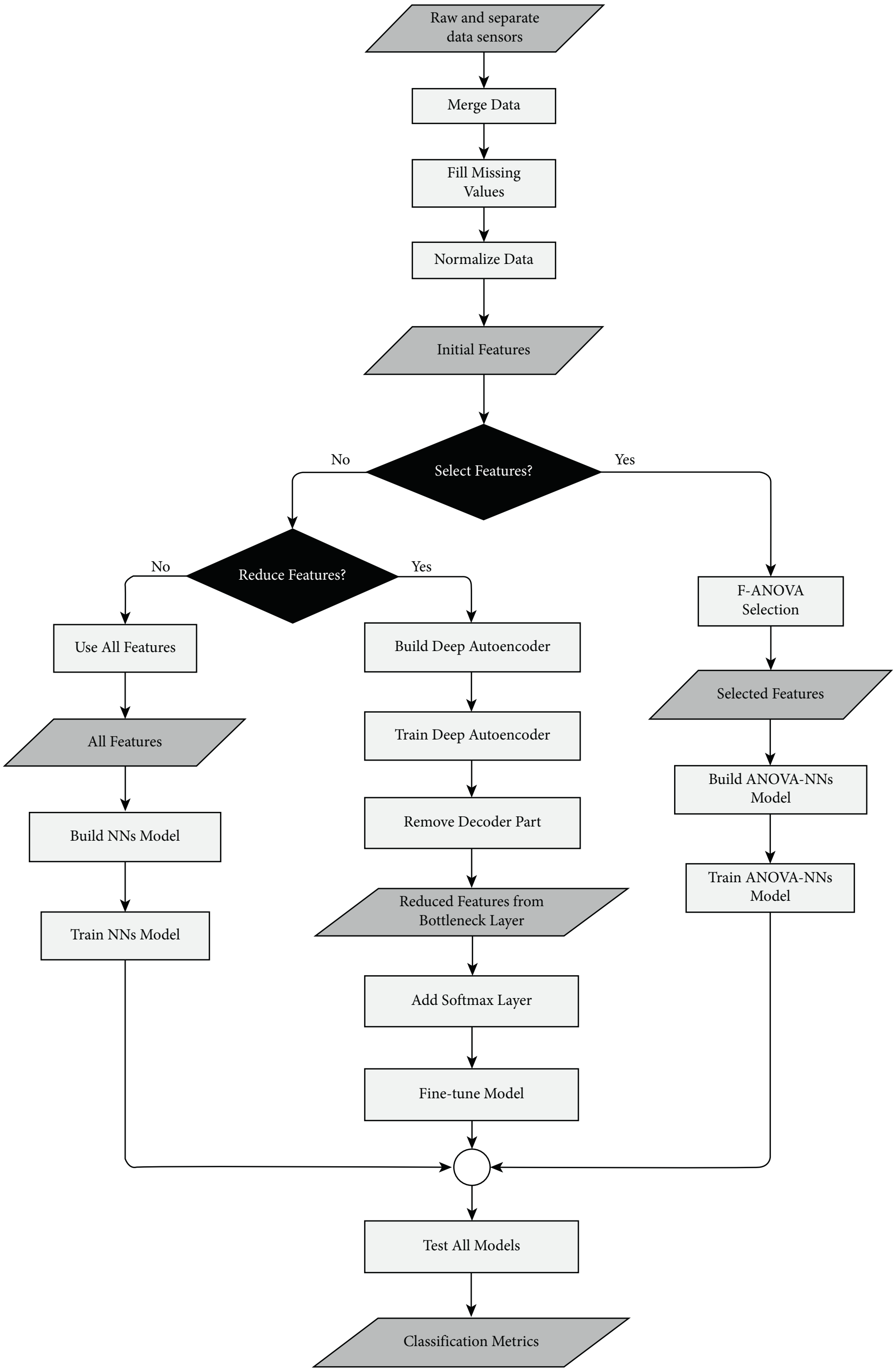

Figure 3: Deep learning-based framework for activity recognition: this framework proposes a detailed methodology for data preprocessing, feature selection and reduction, and training/testing steps for the chosen deep learning techniques.

4.2. Which Features to Use?

4.2.1. The All-Features Approach. Once data processing is complete, we should decide which features to use for the classification task. The first approach is to use all available features without any selection or reduction technique. Despite its basic form, this approach is quite interesting in the deep learning context. In fact, data transformations applied in the intermediate hidden layers can be interpreted as automatic features extraction computations. This process is known as the "Automatic Features Learning" process [56]. These extracted features are repeatedly tuned during the training phase, ending up in optimized and well-suited features for the classification task.

4.2.2. The Feature Selection Approach. The feature selection approach consists of using classic data mining methods to select from the initial number of features a fewer subset of features that are most relevant to the classification task. This approach is widely used by researchers to remove insignificant variables, simplify models, decrease training time, overcome overfitting issues, etc. In this work, we applied, in particular, the "Univariate Selection Approach" also known as the "F-ANOVA selection" (see Figure 3). The key idea of this method is to select the most relevant features by computing univariate statistical tests (ANOVA tests [57]) between each input feature and the target variable. Concretely, if we want k features, we compute ANOVA F-value score between each input and the desired output variable and then we select the best k features having the k highest ANOVA F-value scores.

4.2.3. The Feature Reduction Approach. The feature reduction approach consists of reducing the initial number of features by extracting new feature representations. The extracted features should summarize original data and guarantee to retain a maximum of information. Depending on the used approach, these new features can be a linear or nonlinear transformation of the initial data. In this work, our main approach for dimensionality reduction relies on deep autoencoder architecture. Please refer to Subsection 3.3 for more details on autoencoders. To compare the autoencoder approach, principal component analysis (also known as PCA), a state-of-the-art reduction technique, is also applied. Principal component analysis (PCA) [58] is a statistical approach that aims to map a set of potentially correlated features to a set of linearly uncorrelated features named principal components. Due to an internal mechanism of variance conservation, PCA transforms feature space to a more consistent one that allows reducing the number of features and conserving the original data variability.

4.3. Models Building. In our framework, many models are explored depending on the chosen approach for treating features (see Figure 3). First, for the all-features approach, we simply build the neural network over the complete dataset. The neural network may be a simple MLP, an RNN model, an LSTM model, or a GRU model. These models were particularly chosen since they represent the best state-of-the-art models for activity recognition as pointed by many literature reviews [45, 56]. Please refer to Section 3 for more details about these different neural network architectures. Second, for the feature selection approach, we begin by applying the F-ANOVA selection technique. Next, we build over selected features a neural network model which can be a simple MLP, an RNN model, an LSTM model, or a GRU model. Third, for the reduction approach, we start by building a deep autoencoder using the whole training data. At this step, we remind that the output layer is set to be identical to the input layer. After the training, we remove the decoding part from the AE, keeping only the encoding part. The last layer of the encoding part is the bottleneck layer from which are computed the new features. Indeed, the new reduced features are extracted by getting the output of the bottleneck layer. To finalize the model, we add a "softmax" output layer for the classification task and we retrain the model to fit the new architecture (see Figure 3). This latter training step is known as the fine-tuning step. Finally, all models are tested and evaluated using a multitude of classification metrics. In the next section, we present how our framework was implemented and applied to the Orange4-Home dataset, depending on what feature strategy was used and which model was adopted.

5. Experiments

5.1. Dataset Presentation. For this work, the "Orange4Home dataset" [24] was chosen to test our framework for many reasons. Actually, many datasets already exist in the literature. However, many remarks and limitations are observed on these databases. Opportunity dataset [59] for instance is limited by the presence of only one room, the shortness of recorded sequences (~30 minutes), and the large use of body-worn devices which makes the scenario unreal. Transfer Learning dataset [60] uses 3 houses for experiments but was limited by the number of recording devices and takes into account only 8 classes of activities. ARAS dataset [61] records 27 classes of activities over one whole month in two different smart homes. Activities were labeled in situ by the residents with high accuracy. However, the only use of 20 binary sensors limited the richness of recorded data.

In our work, the "Orange4Home dataset" [24] is chosen due to its numerous advantages and its well-designed protocol. It represents a promising benchmark for behavior studies and routines detection. This dataset is the fruit of a collaboration between INRIA (https://www.inria.fr/en/) and Orange Labs (France). Recordings were effectuated in a two-floor smart apartment, named "Amiqual4Home" (https://amiqual4home.inria.fr/), fully equipped with 236 heterogeneous sensors. Four successive weeks of work (9 hours per day, 5 days per week) were recorded resulting in nearly 180 hours of multimodal data and 24 activity classes labeled accurately in situ by the single resident of the smart house.

The total list of classes is presented in Table 2. Each class is labeled by its location followed by the activity label (e.g., Office|Cleaning). Note that gaps were observed in activity labeling when a transition occurs between two different activities. To deal with this issue, we added a novel class named "inter" activity, which has raised the total number of classes to 25.

Back to the experimental setting, the smart apartment was equipped with several types of ambient sensors generating various types of recorded data as detailed in Table 3 extracted and slightly modified from [24]. Several modalities were sensed ranging from environmental information such as weather variables to more local information such as indoor temperature, noise, doors states, lights states, presence, movements, electric and water consumption, etc. Thanks to the diversity of sensed data and the relevant labeled classes, the "Orange4Home" represents an appropriate benchmark to assess the different components of our framework, namely, the data preprocessing approaches and the deep learning models. In the next subsection, we detail how our

Table 2: List of classes (extracted from [24]). Each class is denoted by its location followed by the activity label.

<table>
<thead>
<tr><th>Entrance|Entering</th><th>Entrance|Leaving</th></tr>
</thead>
<tbody>
<tr><td>Kitchen|Preparing</td><td>Kitchen|Cooking</td></tr>
<tr><td>Kitchen|Washing_the_dishes</td><td>Kitchen|Cleaning</td></tr>
<tr><td>Living_room|Eating</td><td>Living_room|Cleaning</td></tr>
<tr><td>Living_room|Computing</td><td>Living_room|Watching_TV</td></tr>
<tr><td>Staircase|Going_up</td><td>Staircase|Going_down</td></tr>
<tr><td>Office|Computing</td><td>Office|Watching_TV</td></tr>
<tr><td>Office|Cleaning</td><td>Bedroom|Dressing</td></tr>
<tr><td>Bedroom|Reading</td><td>Bedroom|Napping</td></tr>
<tr><td>Bedroom|Cleaning</td><td>Toilet|Using_the_toilet</td></tr>
<tr><td>Bathroom|Showering</td><td>Bathroom|Using_the_sink</td></tr>
<tr><td>Bathroom|Using_the_toilet</td><td>Bathroom|Cleaning</td></tr>
</tbody>
</table>

Table 3: Number of sensors per location and data type. These figures were extracted from [24].

Place\Type	Binary	Numeric	Categorical	Total
Entrance	3	3	3	9
Kitchen	13	39	0	52
Living room	16	14	7	37
Toilet	3	2	0	5
Staircase	3	0	0	3
Walkway	9	1	0	10
Bathroom	9	14	3	26
Office	9	6	5	20
Bedroom	17	10	7	34
Global	1	33	6	40
Total	83	122	31	236

framework was applied to this dataset to get relevant classification results.

5.2. Models Implementation. As previously mentioned, the first component of our framework is data processing. At this level, three steps are applied which are data fusion, filling missing values, and data normalization (see Figure 3). Applying these steps to the Orange4Home dataset results in a feature space composed of 258 normalized features, 25 activity classes, and a total of 224000 timestamp lines. Next, we should decide which features to use. To this end, three approaches were proposed: the first one selects the whole feature space before model building. The second one uses the F-ANOVA technique to select a subset of features, and the third one applies feature reduction techniques. Before presenting all models results, we remind that, for all approaches, 4 neural networks models were tested, namely, MLP, RNN, LSTM, and GRU. The MLP model is built upon two hidden dense layers in addition to input/output layers. The RNN model contains three layers besides input/output layers: one RNN layer followed by one dropout layer (to limit overfitting issues) followed by one dense layer. The same architecture is used for LSTM and GRU models (see Figure 4). This architecture showed to be empirically the most optimal for our dataset. Moreover, for all approaches and all models (MLP, RNN, LSTM, and GRU), the number of nodes in each layer was calculated as follows: (number of inputs + number of classes)/2. This calculation method showed to be empirically the best one for our dataset.

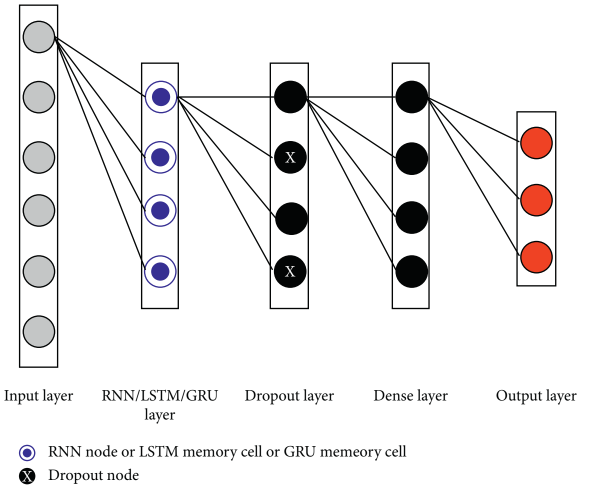

Figure 4: The main topology of neural networks models (RNN, LSTM, and GRU) applied to the Orange4Home dataset.

For the all-features approach, the number of used features for classification corresponds to the initial number of 258 features. For the selection approach, the best classification results were computed using exactly 50 selected features determined by the F-ANOVA technique. Similarly, the number of reduced features was 50 features. In fact, a deep autoencoder model was applied for the reduction approach. Besides input/output layers, the AE model contained three hidden dense layers with, respectively, 100, 50, and 100 nodes (see Figure 2). The input/output layers contained the initial number of features, i.e., 258 nodes. After the training, the decoding part is dropped out, and a “softmax” output layer (with 25 class nodes) is added just after the bottleneck layer, which, we remind, was composed of 50 nodes representing the reduced features. A fine-tune step is lastly computed to retrain the new model parameters. To compare the deep autoencoder results, PCA (principal component analysis) is also tested on our dataset using 50 principal components.

To test all models, 4-cross validation is applied since we have 4 weeks of recording data. In each round, three weeks were used for training and one was used week for testing. Final evaluation metrics represent then the average of the 4 iterations. To conclude this section note that data preprocessing steps, selection techniques, reduction methods, models training, models testing, and assessment metrics were all implemented using Python programming language with the support of well-known Data Science packages such as Numpy, Pandas, Scikit-learn [62], and Keras [63]. All manipulations were ensured by a 16 GB RAM PC characterized by an Intel® Core™ i7-8550U Processor. All results are presented and discussed in the next section.

6. Results and Discussion

6.1. Results of the All-Features Approach. For results comparison, we will essentially rely on two main metrics: the general accuracy and the F-measure which represents the harmonic mean of recall and precision [64]. These evaluation metrics are reminded in the below formulas. For these formulas, note that taking a specific class as a reference, TP refers to True Positive and represents the number of instances correctly assigned to that class. FP refers to False Positive and represents the number of instances incorrectly assigned to that class. FN refers to False Negative and represents the number of instances of that class but assigned to another class. TN (True Negative) refers to the number of instances of another class and correctly assigned to that class.

$$\begin{aligned}\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},\\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\ \text{F} - \text{measure} &= \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}.\end{aligned} \tag{1}$$

In the all-features approach, models were trained using 258 features (the whole feature space). After testing several baseline classic classifiers, the best results (92.22% for accuracy and 92.83% for F-measure) were computed by a random forest classifier [31] (which we denote here as the baseline-RF model). Our proposed MLP model, as well as the RNN model, has succeeded to enhance this result by giving, respectively, 94.26% and 94.20% for the accuracy and 94.73% and 94.74% for the F-measure. The rest of the tested models (i.e., LSTM and GRU) have led to closer results but with no significant amelioration as shown in Table 4. In fact, neural network models have given better results than RF since they have the advantage of the "automatic features learning" process. As explained before, extracted features from this process are repeatedly tuned during the training phase, ending up in optimized and well-suited features for the classification task. Moreover, looking at the computing times in Table 4, neural networks models have recorded, as expected, longer times than RF due to the complexity of these models. Besides having the best rates, MLP and RNN models present also better computing times compared to LSTM and GRU. For instance, for the testing time, the RNN model has given 3.71 sec versus 5.48 sec for the LSTM. Therefore, both MLP and RNN models present a good tradeoff taking into account all evaluation metrics in the all-features approach.

6.2. Results of the Feature Selection Approach. Similar to the all-features approach, after testing several baseline classic classifiers, the feature selection approach has given optimal results using a random forest classifier. Thanks to the F-ANOVA technique, the optimal number of selected features was 50 features, the accuracy rate was equal to 95.10%, and the F-measure was equal to 95.30% as shown in Table 5. Next, the same selection technique was coupled with our framework classifiers (MLP, RNN, LSTM, and GRU) resulting in the following new models: ANOVA-MLP, ANOVA-RNN, ANOVA-LSTM, and ANOVA-GRU. The baseline-RF model is denoted here as the baseline-ANOVA-RF. While a slight accuracy amelioration was recorded for the neural network models (especially 95.37% for ANOVA-MLP versus 95.10% for ANOVA-RF), the F-measure scores were very close for all models. For instance, the ANOVA-MLP classifier has given an F-score of 95.39% with no significant difference with ANOVA-RF results. The same finding was observed for the rest of the tested models (i.e., ANOVA-RNN, ANOVA-LSTM, and ANOVA-GRU). In fact, for neural networks, the internal process of automatic feature selection has given these models some advantage over the RF model in the all-features approach since no selection operation was applied at all. When applying the F-ANOVA selection technique, the internal process of feature selection seems to be unbeneficial for NNs models since here all classifiers share the same selection approach. Furthermore, the comparison between these two approaches (all-features and the selection one) shows that selection techniques have succeeded to improve classification performances for overall models. As a consequence, when building smart houses, it is not necessary to have full instrumentation and a high number of sensors for efficient activity detection. It is rather more optimal to precede with a smart choice of sensors number, sensors type, and their locations. This approach presents many advantages for reducing costs and technical issues and especially helps in resolving many challenges related to resident's privacy since fewer sensors are needed to capture their movements. Last, for the computing times, similar results to the previous subsection were observed since recorded times were equivalent to the complexity of models; i.e., the more complex the model is, the longer the computing time is.

6.3. Results of the Feature Reduction Approach. In this work, our main approach for feature reduction is to apply the autoencoder method as described in Subsection 5.2. The resulting model for the classification task is denoted here as the AE-MLP model. But before exposing this model's results, a baseline model based on an RF classifier coupled to a PCA reduction strategy (50 components from 258 features) has given a rate of 79.68% for accuracy and 81.30% for F-measure. This baseline model is denoted here as the baseline-PCA-RF model. Next, we have tested a PCA-MLP approach by combining a PCA reduction technique with an MLP classifier. Results were significantly enhanced from 79.68% to 86.58% for accuracy and from 81.30% to 88.04% for F-measure as shown in Table 6. Afterward, we tested our main proposed approach (the AE-MLP model) and results were significantly improved to reach an accuracy of 94.09% and an F-measure of 94.26% as shown in Table 6. Therefore, our proposed AE-MLP model has outperformed all other reduction models with relevant accuracy and F-measure classification rates. Moreover, the computing times of the AE-MLP model were close to the PCA-MLP model which supports the relevance of our proposed approach. This result confirms previous findings in literature showing advantages

Table 4: Results of all models in the "all-features approach."

All-features approach (258 features)	Accuracy (%)	F-measure (%)	Training time (sec)	Testing time (sec)
Baseline-RF	92.22	92.83	12.21	0.96
MLP	**94.26**	**94.73**	61.99	3.13
RNN	**94.20**	**94.74**	84.83	3.71
LSTM	93.53	94.27	153.06	5.48
GRU	93.40	94.31	153.94	5.13

Table 5: Results of all models in the "feature selection approach."

Feature selection approach (50 features)	Accuracy (%)	F-measure (%)	Training time (sec)	Testing time (sec)
Baseline-ANOVA-RF	95.10	95.30	4.94	0.83
ANOVA-MLP	**95.37**	**95.39**	46.3	3.08
ANOVA-RNN	95.21	95.27	106.71	2.96
ANOVA-LSTM	95.27	95.30	135.04	3.62
ANOVA-GRU	95.14	95.20	142.87	3.67

Table 6: Results for the "feature reduction approach."

Feature reduction approach (50 reduced features)	Accuracy (%)	F-measure (%)	Training time (sec)	Testing time (sec)
Baseline-PCA-RF	79.68	81.30	112.56	0.61
PCA-MLP	86.58	88.04	41.94	3.29
AE-MLP	**94.09**	**94.26**	54.61	4.15

of autoencoders over PCA since autoencoders perform nonlinear transformations when compacting reduced features, contrary to PCA which performs simple linear calculations. Another advantage of the AE-MLP approach is the homogenous architecture of this model. In fact, both reduction and classification models are based on neural network layers and then combined to a single deep neural network model. This homogenous topology enables optimal training and fine-tuning at the same time of the two parts of the model, namely, the reduction part and the classification part.

7. Conclusion

\In this paper, we have proposed a deep learning-based framework to decide the best approaches for ADL classification in smart home. The framework was intended to ensure relevant data preprocessing steps, test all possible strategies to choose relevant sensors, and apply the best state-of-the-art models for classification. In fact, after preprocessing, three approaches were explored to determine optimal features, namely, the all-features approach, the selection feature approach, and the reduction feature approach. The F-ANOVA technique is applied for the selection approach, while a deep autoencoder (AE) is proposed as the main technique for the reduction approach. The last component of the proposed framework is the ADL recognition models. Four neural network models were chosen thanks to their proven discriminative power which are MLP, RNN, LSTM, and GRU. Our new framework was applied to the "Orange4Home" dataset which represents a promising dataset for smart homes research. As a result, our framework has shown significant improvement over baseline models computed on the same dataset. Actually, for the all-features approach, our RNN-based model has given the best F-measure rate (94.74%) compared to the baseline-RF classifier (92.83%). For the reduction approach, our AE-MLP model has largely succeeded to outperform the baseline-PCA-RF model (94.26% vs. 81.30%), thanks to its consistent architecture and training process. Furthermore, the best overall results were given by the feature selection approach suggesting that heavy installation settings in smart homes may not lead to optimal performances for ADL recognition. As a consequence, it may be more beneficial to limit the number of sensors and to choose smartly their types and locations. This recommendation represents a relevant approach to tackle many smart homes concerns related to interactivity, costs, and privacy. To conclude, we would like to emphasize that our proposed framework is designed to be easily generalized and applicable to any smart home dataset equipped with ambient sensors. It can be a useful and powerful guideline for researchers and engineers to process any ADL classification task. For the perspectives of this research, we intend to develop the framework to include other types of sensors such as wearable devices, audio sensors, and video-based sensors. Moreover, many extensions of this framework are studied to ameliorate recognition rates using incremental learning techniques and deep reinforcement learning approaches.

Data Availability

The author has used third-party data. More information about these data can be obtained from the reference Cumin J., Lefebvre G., Ramparany F., Crowley J.L. (2017) A Dataset of Routine Daily Activities in an Instrumented Home; Ochoa S., Singh P., Bravo *J.* (eds) Ubiquitous Computing and Ambient Intelligence. UCAmI 2017; Lecture Notes in

Computer Science, vol 10586. Springer, Cham. https://doi.org/10.1007/978-3-319-67585-5_43.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] A. Gustavsson, L. Jonsson, T. Rapp et al., "Differences in resource use and costs of dementia care between European countries: baseline data from the ictus study," *The Journal of Nutrition, Health & Aging*, vol. 14, no. 8, pp. 648–654, 2010.

[2] R. J. Robles and T. Kim, "Applications, systems and methods in smart home technology: a review," *International Journal of Advanced Science and Technology*, vol. 15, p. 12, 2010.

[3] M. P. Lawton and E. M. Brody, "Assessment of older people: self-maintaining and instrumental activities of daily living," *The Gerontologist*, vol. 9, pp. 179–186, 1969.

[4] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring activities of daily living in smart homes: understanding human behavior," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 81–94, 2016.

[5] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1662–1674, 2013.

[6] A. Benmansour, A. Bouchachia, and M. Feham, "Multioccupant activity recognition in pervasive smart home environments," *ACM Computing Surveys*, vol. 48, no. 3, pp. 1–36, 2016.

[7] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, "Smart homes and their users: a systematic analysis and key challenges," *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 463–476, 2015.

[8] L. Piyathilaka and S. Kodagoda, "Human activity recognition for domestic robots," in *Field and Service Robotics: Results of the 9th International Conference*, L. Mejias, P. Corke, and J. Roberts, Eds., Springer International Publishing, Cham, Switzerland, pp. 395–408, 2015.

[9] H. Admoni and B. Scassellati, "Data-driven model of nonverbal behavior for socially assistive human-robot interactions," in *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 196–199, New York, NY, USA, November 2014.

[10] A. Mihoub and G. Lefebvre, "Wearables and social signal processing for smarter public presentations," *ACM Transactions on Interactive Intelligent Systems*, vol. 9, pp. 1–24, 2019.

[11] T. Gan, J. Li, Y. Wong, and M. S. Kankanhalli, "A multi-sensor framework for personal presentation analytics," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, pp. 1–21, 2019.

[12] A. Mihoub and G. Lefebvre, "Social intelligence modeling using wearable devices," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pp. 331–341, New York, NY, USA, March 2017.

[13] D. Ravì, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 56–64, 2017.

[14] B. Barshan and M. C. Yuksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Computer Journal*, vol. 57, no. 11, pp. 1649–1667, 2014.

[15] A. Jalal and M. Mahmood, "Students' behavior mining in e-learning environment using cognitive processes with information technologies," *Education and Information Technologies*, vol. 24, no. 5, pp. 2797–2821, 2019.

[16] A. Mihoub, G. Bailly, C. Wolf, and F. Elisei, "Graphical models for social behavior modeling in face-to face interaction," *Pattern Recognition Letters*, vol. 74, pp. 82–89, 2016.

[17] A. Mihoub, G. Bailly, C. Wolf, and F. Elisei, "Learning multimodal behavioral models for face-to-face social interaction," *Journal on Multimodal User Interfaces*, vol. 9, pp. 1–16, 2015.

[18] Z. Zhu, H. F. Satizábal, U. Blanke, A. Perez-Uribe, and G. Tröster, "Naturalistic recognition of activities and mood using wearable electronics," *IEEE Transactions on Affective Computing*, vol. 7, no. 3, pp. 272–285, 2016.

[19] A. Mihoub, G. Bailly, and C. Wolf, "Social behavior modeling based on incremental discrete hidden Markov models," in *Proceedings of the Human Behavior Understanding*, pp. 172–183, Barcelona, Spain, 2013.

[20] A. Mihoub, G. Bailly, and C. Wolf, "Modeling perception-action loops: comparing sequential models with frame-based classifiers," in *Proceedings of the Second International Conference on Human-Agent Interaction*, pp. 309–314, New York, NY, USA, October 2014.

[21] R. Chavarriaga, H. Sagha, A. Calatroni et al., "The opportunity challenge: a benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[22] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[23] T. Nef, P. Urwyler, M. Büchler et al., "Evaluation of three state-of-the-art classifiers for recognition of activities of daily living from smart home ambient data," *Sensors*, vol. 15, no. 5, pp. 11725–11740, 2015.

[24] J. Cumin, G. Lefebvre, F. Ramparany, and J. L. Crowley, "A dataset of routine daily activities in an instrumented home," in *Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence*, pp. 413–425, Philadelphia, PA, USA, November 2017.

[25] U. A. B. U. A. Bakar, H. Ghayvat, S. F. Hasanm, and S. C. Mukhopadhyay, "Activity and anomaly detection in smart home: a survey," in *Next Generation Sensors and Systems*, S. C. Mukhopadhyay, Ed., Springer International Publishing, Cham, Switzerland, pp. 191–220, 2016.

[26] D. Singh and C. Krishna Mohan, "Graph formulation of video activities for abnormal activity recognition," *Pattern Recognition*, vol. 65, pp. 265–272, 2017.

[27] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, Springer International Publishing, Cham, Switzerland, 2015.

[28] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[29] S. Suthaharan, "Support vector machine, machine learning models and algorithms for big data classification.," in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, S. Suthaharan, Ed., Springer US, Boston, MA, USA, pp. 207–235, 2016.

[30] H. Fang, P. Tang, and H. Si, "Feature selections using minimal redundancy maximal relevance algorithm for human activity recognition in smart home environments," *Journal of Healthcare Engineering*, vol. 2020, Article ID 8876782, 13 pages, 2020.

[31] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] M. Jethanandani, A. Sharma, T. Perumal, and J.-R. Chang, “Multi-label classification based ensemble learning for human activity recognition in smart home,” *Internet of Things*, vol. 12, Article ID 100324, 2020.

[33] O. Brdiczka, M. Langet, J. Maisonnasse, and J. L. Crowley, “Detecting human behavior models from multimodal observation in a smart home,” *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 4, pp. 588–597, 2009.

[34] L. Nguyen, “Tutorial on hidden Markov model,” *Applied and Computational Mathematics*, vol. 24, 2016.

[35] P. Asghari, E. Soleimani, and E. Nazerfard, “Online human activity recognition employing hierarchical hidden Markov models,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1141–1152, 2020.

[36] M. M. Hassan, S. Huda, M. Z. Uddin, A. Almogren, and M. Alrubaian, “Human activity recognition from body sensor data using deep learning,” *Journal of Medical Systems*, vol. 42, no. 6, p. 99, 2018.

[37] D. Arifoglu and A. Bouchachia, “Activity recognition and abnormal behaviour detection with recurrent neural networks,” *Procedia Computer Science*, vol. 110, pp. 86–93, 2017.

[38] D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist, and A. Holzinger, “Convolutional and recurrent neural networks for activity recognition in smart environment,” in *Towards Integrative Machine Learning and Knowledge Extraction*, A. Holzinger, R. Goebel, M. Ferri, and V. Palade, Eds., Springer International Publishing, Cham, Switzerland, pp. 194–205, 2017.

[39] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, M. Ryan, M. Elhoseny, and H. Song, “ST-DeepHAR: deep learning model for human activity recognition in IoHT applications,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4969–4979, 2021.

[40] A. A. Alani, G. Cosma, and A. Taherkhani, “Classifying imbalanced multi-modal sensor data for human activity recognition in a smart home using deep learning,” in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Glasgow, UK, July 2020.

[41] H. Yang, S. Gong, Y. Liu, Z. Lin, and Y. Qu, “A multi-task learning model for daily activity forecast in smart home,” *Sensors*, vol. 20, no. 7, p. 1933, 2020.

[42] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[43] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, July 2015.

[44] S. Ramasamy Ramamurthy and N. Roy, “Recent trends in machine learning for human activity recognition-A survey,” *WIREs Data Mining Knowl Discov*, vol. 8, p. e1254, 2018.

[45] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: a survey,” *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[46] O. Ben Fredj, A. Mihoub, M. Krichen, O. Cheikhrouhou, and A. Derhab, “CyberSecurity attack prediction: a deep learning approach,” in *Proceedings of the 13th International Conference on Security of Information and Networks*, New York, NY, USA, November 2020.

[47] S. Thomas, M. Bourobou, and J. Li, “Ensemble of deep autoencoder classifiers for activity recognition based on sensor modalities in smart homes,” in *Data Science*, Q. Zhou, Q. Miao, H. Wang, W. Xie, Y. Wang, and Z. Lu, Eds., Springer, Singapore, pp. 273–295, 2018.

[48] A. Wang, G. Chen, C. Shang, M. Zhang, and L. Liu, “Human activity recognition in a smart home environment with stacked denoising autoencoders,” in *Web-Age Information Management*, S. Song and Y. Tong, Eds., Springer International Publishing, Cham, Switzerland, pp. 29–40, 2016.

[49] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, “DeepSense: device-free human activity recognition via autoencoder long-term recurrent convolutional network,” in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, London, UK, June 2018.

[50] A. K. Jain, J. Jianchang Mao, and K. M. Mohiuddin, “Artificial neural networks: a tutorial,” *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[51] S. Osowski, K. Siwek, and T. Markiewicz, “MLP and SVM networks–a comparative study,” in *Proceedings of the Nordic Signal Processing Symposium*, Espoo, Finland, June 2004.

[52] Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, and M. S. Nasrin, “The history began from AlexNet: a comprehensive survey on deep learning approaches,” 2018, https://arxiv.org/abs/1803.01164.

[53] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, 1998.

[54] Y. Guan and T. Plötz, “Ensembles of deep LSTM learners for activity recognition using wearables,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, 2017.

[55] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[56] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges,” *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.

[57] R. G. M. Jr. Rupert, *Beyond ANOVA: Basics of Applied Statistics*, Chapman and Hall/CRC, Boca Raton, FL, USA, 1997.

[58] I. T. Jolliffe, *Principal Component Analysis*, Springer Science & Business Media, Berlin, Germany, 2006.

[59] D. Roggen, A. Calatroni, M. Rossi et al., “Collecting complex activity datasets in highly rich networked sensor environments,” in *Proceedings of the 2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pp. 233–240, Kassel, Germany, June 2010.

[60] T. L. M. Van Kasteren, G. Englebienne, and B. J. A. Kröse, “Transferring knowledge of activity recognition across sensor networks,” in *Proceedings of the International Conference on Pervasive Computing*, pp. 283–300, Toronto, Canada, March 2010.

[61] H. Alemdar, H. Ertan, O. D. Incel, and C. Ersoy, “ARAS human activity datasets in multiple homes with multiple residents,” in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, pp. 232–235, Brussels, Belgium, May 2013.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort et al., “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[63] F. Chollet, *Others*, Keras, London, UK, 2015.

[64] D. M. Powers, *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation*, 2011, https://arxiv.org/abs/2010.16061.