

# Recognizing Activities of Daily Living using 1D Convolutional Neural Networks for Efficient Smart Homes

Sumaya Alghamdi<sup>1</sup>, Etimad Fadel<sup>2</sup>, Nahid Alowidi<sup>3</sup>  
Faculty of Computing and Information Technology  
King Abdulaziz University, Jeddah, KSA

**Abstract**—Human activity recognition is considered a challenging task in sensor-based monitoring systems. In ambient intelligent environments, such as smart homes, collecting data from ambient sensors is useful for recognizing activities of daily living, which can then be used to provide assistance to inhabitants. Activities of daily living are composed of complex multivariable time series data that has high dimensionality, is huge in size, and is updated constantly. Thus, developing methods for analyzing time series data to extract meaningful features and specific characteristics would help solve the problem of human activity recognition. Based on the noticeable success of deep learning in the field of time series classification, we developed a model called a deep one-dimensional convolutional neural network (Deep 1d-CNN) for recognizing activities of daily living in smart homes. Our model contains several one-dimensional convolution layers coupled with max-pooling technique to learn the internal representation of time series data and automatically generate very deep features for recognizing different activity types. For the performance evaluation, we tested our deep model on the new real-life dataset, ContextAct@A4H, and the results showed that our model achieved a high F1 score (0.90). We also extended our study to show the potential energy saving in smart homes through recognizing activities of daily living. We built a recommendation system based on the activities recognized by our deep model to detect the devices that are wasting energy, and recommend the user to execute energy optimization actions. The experiment indicated that recognizing activities of daily living can result in energy savings of around 50%.

**Keywords**—Deep learning; one-dimensional convolutional neural networks; time-series classification; Activities of Daily Living (ADLs); smart home; recommendation system

## I. INTRODUCTION

In recent decades, internet of things (IoT) applications have been increasing in popularity, and one well-known application is smart homes. These environments relied using sensors to generate large amounts of time series data, which can be analyzed for many purposes, such as monitoring and detecting activities in order to make timely decisions [1]. Recognizing human activities in an automated manner has become essential in many ambient intelligence applications, and a smart home with artificial intelligence (AI) is a great opportunity towards it [2]. Human activity recognition (HAR) in smart homes can be categorized into sensor-based activity recognition and vision-based activity recognition. Sensor-based activity recognition

can be done using two kinds of sensors: wearable sensors, which are placed on a human body to capture frequent motions, such as walking and standing, and ambient sensors, which are used to recognize activities of daily living (ADLs), such as watching television, and cooking. The main difference between these two kinds of sensors is that sensor-based ambient systems have the ability to recognize different behaviors and complex activities by monitoring the interaction between objects and inhabitants in smart homes, while wearable sensors can only capture human physical movements [3]. Recently, increased attention has been given to deep learning in several fields, including speech recognition [4] and image classification [5]. In HAR, deep learning can have a great impact in terms of system performance and flexibility. Deep learning methods can provide efficient tools for high-level feature extraction from multivariable time series data, which is beneficial for many tasks, such as classification. Furthermore, deep learning models have demonstrated significant potential to outperform the state of the art in HAR. They decrease the need for extracting features manually which is usually a time-consuming task and can be error-prone or (at least) a poorly generalizable endeavor. In general, deep learning involves neural networks comprised of several layers that process non-linear information organized hierarchically, where the output of each layer is the input for the next one. Well-known deep learning architectures for time series classification include one-dimensional convolutional neural networks (1d-CNNs) [6] and recurrent neural networks (RNNs) [7]. Since the multivariable time series data of ADLs is high-dimensional data with significant variation, applying feature selection methods on the data before feeding it into a deep learning network would help reduce the complexity of activity recognition and increase the performance of such a system. Indeed, feature selection is a substantial step when dealing with multivariable time series data, allowing for the elimination of variables that are not useful or irrelevant for the model description [8]. In addition, deep learning models (i.e., 1d-CNNs) have been shown to be capable for classifying time series data by automatically capturing local dependencies and preserving feature scale invariance to catch variation in the same class throughout the feature extraction [9]. Thus, combining a feature selection technique with 1d-CNN will result in highly accurate system for recognizing ADLs, which can be then used as a basis for decision making processes, such as those related to energy saving decisions. In fact, recognizing ADLs can increase the energy saving potential of a smart home by using sensory

readings to detect the appliances that are wasting energy and recommend the user to perform energy optimization actions based on the current performed activity [10]. Therefore, daily activity recognition and a learning process in smart homes are necessary and important.

In this work, we take advantage of a feature selection method combined with 1d-CNN to accurately perform activity recognition since, in general, CNNs are good at reducing frequency variations and capturing local dependency. Once our system detects the current performed activity, energy saving recommendations are generated to inform the user of any energy optimization actions. We explored our framework, including the deep model and the recommendation system, on a new large dataset entitled ContextAct@A4H [11] to determine the capacity for activity recognition and study the energy saving potential of recognizing ADLs.

## II. RELATED WORK

In this section, we discuss recent research applying deep learning networks to recognize ADLs in ambient-based environments (i.e., smart homes). Then, we review some studies that built model-based activity recognition to increase the energy saving potential in smart homes.

### A. Activity Recognition using Deep Learning-based Approaches

In the HAR field, the use of sensors is extremely diverse. As a matter of fact, activity recognition is used in wearable devices to recognize human body movements, such as walking and running, and in ambient sensors to recognize complex daily activities such as taking shower [3]. Among the various activity recognition fields, we are focused on recent studies of daily activity recognition-based ambient intelligent systems, specifically, smart homes. This process can be done by applying ML approaches, which can effectively classify ADLs performed by people. Traditional ML-based approaches are popular in this field, and most of them are supervised learning approaches. Although classical techniques, such as SVM, are simple to use, they require manual feature extraction from the data, which is time consuming and heavily dependent on human knowledge of the domain [12]. Recently, there has been a shift in the use of deep learning techniques in activity recognition to outdo the ML traditional methods by overcoming the need for feature engineering. Many researchers have demonstrated the significant potential of deep learning methods to push the state of the art in HAR. The most frequently used deep learning methods for recognizing ADLs are 1d-CNN, Vanilla RNN, long short-term memory (LSTM), and gated recurrent unit (GRU) [13]. These deep learning methods are summarized in Table I.

In general, the input of the neural network is original raw data, and extracting features from this data tends to maximize the overall performance. Extracting deep information from time series data requires expert knowledge; however, CNNs have been proposed to handle this problem. Therefore, CNNs are one of the most used deep learning approaches in the HAR field not only for extracting features from data but also for activity classification [14]. In ambient-based environments, CNNs have been proposed to extract features from time series

data optioned from ambient sensors for daily activity classification. We introduce these works below.

Recently, [15] exploited CNN to perform activity recognition in smart homes. Specifically, the classification of ADLs, such as cooking and sleeping, were performed using a temporal 1d-CNN model. The proposed deep learning approach contained the input layer, the multiple one-dimensional convolution layers, the fully connected layer, and the softmax classifier. They used 128 filters for each convolution layer with different 1d kernel size. The researchers trained the model with a one-day-left strategy. For the evaluation, they tested their model on the publicly available dataset (Kasteren) and they compared the obtained results with other probabilistic methods, such as naive Bayes (NB) and conditional random fields (CRFs). Their results stated that the 1d-CNN model showed significant improvement in performance in terms of accuracy compared to other classifiers.

A comparison was conducted in [16] to evaluate state of the art ML techniques and the deep learning models for the classification of ADLs in smart homes, including CNNs, LSTM, SVM, and the hidden Markov model (HMM). They used a synthetic dataset for real-life activities and a real-world dataset, which is known as ARAS, to evaluate and compare the performance of the traditional ML techniques and the deep learning models. The performance metrics used for the evaluation were precision, f1 score, and recall metrics. The results of the comparison indicated that the CNN model had the best performance compared to the other evaluated techniques across both datasets in term of F1 score. Since the ADL data consists of sequences recorded by sensors, models based on RNNs have been widely proposed to solve the complexity of sequence lapping problems in daily activity recognition.

A model based RNN was proposed by [17] to learn how to classify daily activities without any prior knowledge. They applied LSTM and evaluated its performance on a publicly available and annotated dataset (Kasteren). For the evaluation, they left one day out of the training and used that data for the testing phase. They compared the performance of their LSTM model and the following common classifiers: NB, HMM, Hidden Semi-Markov model (HSMM). The proposed LSTM model showed a 40% improvement in accuracy compared to the other classifiers.

More recently, paper [18] used vanilla RNN in a learning approach for HAR since human activities generate sequences that contain time-dependent sensor records. Before the training the RNN model phase, they applied dynamic windowing approach in which each window contains the best fitting sensor set for each activity; the obtained windows are then feed the RNN classification model. The proposed RNN model had eight neurons and was trained using a back-propagation algorithm. Their proposed RNN model was testing using 10-fold cross validation. To evaluate their approach, they used three popular datasets for daily activities: Towor, Aruba, and HBMS. The results showed a high overall classification performance for their proposed model based on different evaluation metrics and compared with common classification models, such as SVM and K-Nearest Neighbor (KNN).

TABLE I. THE DEEP LEARNING APPROACHES MOST FREQUENTLY USED FOR RECOGNIZING ADLS

Disadvantage	Advantage	Description	Methods
1d-CNN	It is a multilayer neural network that combines convolution, activation, and pooling layers.	It automatically extracts very deep and discriminative features from sequences of sensory data.	It requires a big dataset and a high number of hyper-parameter tuning to achieve optimal good performance in activity recognition.
Vanilla RNN	It is the classical version of RNN, and it has a temporal layer used to capture sequential information.	It performs well at modelling sequence data and learning changes over time.	It is not good at capturing long-term dependencies on sequences due to the vanishing gradient problem.
LSTM	It is a modern version of RNN that has a memory block (cell state) with many gates to overcome the vanishing gradient problem.	It can capture long temporal dependencies to enable modelling of complex activity details.	LSTMs require too many parameters to be updated during the training time.
GRU	It is an RNN-based model with fewer gates to detect and recognize time complex events.	Same as LSTM except it has fewer parameters and is easy to train.	It only shows good performance on small data.

Meanwhile, [19] studied the problem of activity recognition and abnormal behavior detection among older people with dementia. They investigated three RNN models: LSTM, Vanilla RNN, and GRU. They considered activity recognition to be a sequence labeling problem, making it appropriate for RNNs. To assess RNNs in the activity detection, they followed many steps. First, they segmented the raw dataset into windows by applying the sliding window approach. Then, they fed those instances and their corresponding labels into the RNN-based models to train them to recognize daily activities. To fulfill their purpose, they used a popular dataset gathered by Van Kasteren from three houses to train their models. They divided the dataset into training and testing sets, applying the leave one day out cross-validation approach. To provide an adequate discussion of the performance of the RNN-based models (LSTM, Vanilla RNN, and GRU), they compared them against state-of-the-art classifiers, including SVM, HMM, SHMM, and CRF. Their results showed that the RNNs were very competitive with CRF and much better than the other traditional ML methods.

### B. Energy Efficient Smart Home-based Activity Recognition System

In the past decade, researchers have integrated ADLS into the energy management domain in order to increase the energy efficiency of smart homes. Recognizing ADLS could be an excellent way to obtain insights into occupant behavior, which are required for detecting potential energy savings at the smart home [20].

In general, there are two kinds of energy saving systems: the active variant, which directly controls the devices in the smart home, and the passive variant, which is used to increase the awareness of the resident by notify them about the possible energy saving actions [21]. Through the literature, both systems have been integrated in many studies with the process of recognizing ADLS to enhance the potential energy saving in the smart home; we review these works below.

To begin with, paper [10] designed a method called Activity-Appliance-Energy Consumption (AAEC) for energy saving in smart homes based on the relationship between electrical appliances and user activities. In this method, user activity was recognized by analyzing the information gathered from sensors about appliances and how the user interacted with them. The process of activity recognition was performed using two ML techniques: Random Tree (RT) and Logistic

Regression (LR). After detecting the activity, a ranking algorithm was utilized to find a relationship between each activity and the home appliances. Decision Tree (DR) algorithm was then used to define how important each specific appliance is for an activity. Finally, based on the relationship and the importance level between the current performed activity and the appliances, the system sent recommendations to the user whenever it detects a waste of energy. To validate their proposed method, the authors used data obtained from an MIT project. The result showed an average energy savings of about 35%, which was accomplished by applying the AACE method in the smart environment.

Based on the idea that most ADLS at home are related to a number of electrical appliances, researchers in [21] developed a framework called SMARTENERGY.KOM, which aims to achieve energy efficiency in buildings based on recognizing ADLS. For activity recognition, the authors showed how it was possible to recognize user activity by monitoring the energy consumption of appliances to detect which ones were running. To make their activity detections more accurate, they monitored additional information, such as temperature, brightness, and motion in the smart home. The activity recognition performed using random forest classifier. For energy saving, they designed the EnergyAdvisor, which was responsible for directly generating energy saving recommendations once it detected any appliances that were turned on but were not related to the user's current performed activity. The relationship between the activities and the appliances was defined by setting a set of static rules. Using the data they gathered, the authors demonstrated that, by combining power sensors with information obtained from an environment sensor, their system was capable of detecting activity with high level of accuracy and significantly able to save energy.

One work conducted by [22] proposed an activity-aware smart automation system called CASAS activity aware resource learning (CARL). The aim of CARL is to automate a smart building by turning off appliances that are not needed for the current performed activity or any activities that are anticipated to occur within the subsequent 10 minutes. The hypothesis of this work was smart buildings with energy-efficient building automation based activity recognition could realize energy savings while still meeting the needs of the individuals who live there. In their building automation approach, activity learning played two roles. First, activity recognition was applied to identify activities as they were

performed in a smart building environment using SVM. Second, activity prediction was used to predict whether a particular activity would happen within a specific upcoming period of time. Together, these roles provided a basis for building automation that supported the current and upcoming tasks of the residents in the building. CARL's goal was to turn off many devices without interfering with resident tasks. Performance can thus be evaluated in terms of the number of times a device is turned off (to measure energy consumption) and the number of times a user double tapped a switch in order to provide feedback to the system while turning the device back on (to evaluate the accuracy of the event generated). The experiments were run on a smart apartment named Navan, a single-resident apartment with a floor plan equipped with 118 sensors. Using CARAL resulted in reducing device utilization by 56% and an energy savings of up to 50%.

An algorithm based on HMM was proposed by [23] to recognize the activity states of smart home inhabitants and then control the operation of their appliances accordingly. The researchers designed a smart home energy management system (SHEMS), which was based on deploying motion sensors to collect information and actuators to control the appliances in the home. The SHEMS was able to recognize three state of activities: active, sleep, and away. In the active state, the user was at home, and he performed some tasks. In the sleep state, the user was at home but not active, and the away state detected when the user was not at home. Based on the recognized state, the appliances were controlled according to a pre-defined schedule. Each state was identified by monitoring the motion inside the home and the logical status of the main door. HMM was used to estimate the portability for each state to determine which one was occurring. Based on two scenarios, the system was evaluated successfully. The results indicated that the SHEMS could lead to an energy savings of about 18%.

Recently, a work done in [24] aims to recognize daily life activities performed by inhabitants in a smart home in order to save more energy by guaranteeing that the peak demands won't exceed a pre-defined threshold. Their proposed system was able to convert sensory data into meaningful event using a lossy compression algorithm based on minimum description length. Then, from the event sequence, patterns representing activities were extracted and clustered to train a HMM to automatically recognizing activities. Moreover, for each activity, there was a typical consumption model associated with each one identified from the involved appliances. Based on the current detected activity, the system could guarantee that energy consumption won't exceed a predefine threshold to prevent the arising of unwanted peak. Their approach based on delaying the activation of some appliances which were not directly required in the current activity till the demand of energy getting below the giving threshold. Their experiment was performed on three public datasets for ADLs and the results demonstrated that their approach was able to save energy of about 30.

Based on the aforementioned works, we can see that it is possible to design a system that can suggest actions in smart homes based on recognizing ADLs, lowering energy usage without decreasing the comfort levels of the inhabitants. Our work aimed to design deep networks that automatically

recognize daily activities in an ambient-based environment. Model-based RNNs have been widely applied for time series classification. However, those models are hard to train and require a large memory and a long time to update their parameters in each cell state. Moreover, RNN-based models are usually good at predicting what comes next in a sequence, while CNNs can learn to classify a sequence. Thus, building a 1d-CNN to classify time series data for detecting ADLs in ambient-based environment seems to be more logical than building an RNN based model. CNNs can effectively extract very deep features that represent each activity and can be trained faster than RNN-based models with the same or even better results, as shown by [16]. Using the CNN-based model to classify time series data for activity recognition will lead to significant improvements in performance, increase accuracy of the system and provide better results. Moreover, developing a recommendation system that informs inhabitants about potential energy saving actions through ADLs will make their home more efficient, as well as positively will influence their behaviors and make them more aware toward the energy.

### III. OVERALL DESIGN

#### A. Overview

Before analyzing our data, data preprocessing steps were applied. First, data cleaning was implemented, followed by data normalization and splitting (see section 4, subsection B for more detail). Then, a feature selection method was applied to identify the related features from the dataset and remove the less important features (see Section 4, subsection C for more details). After that, a segmentation-based approach was used to obtain sequences from the ambient sensor data in order to feed the input of our deep model. The time series segmentation method we applied was a fixed-sized overlapping sliding window that divided the data into multiple sequences of discrete segments that shared the same length and were assigned to different labels (see Section 4, subsection D for more details). After making the data suitable for analysis, a deep learning model (Deep 1d-CNN) was proposed to classify the time series data for recognizing ADLs. Then, the output of Deep 1d-CNN fed the recommendation system, which was responsible for generating energy saving recommendations to the user. These recommendations were generated when the system detected a waste of energy based on the current performed activity using predefined rules based on the relationships between the activity and the selected energy sensors.

The two main objectives of our work, which were unified in one framework, were: 1) to provide a high level of accuracy in recognizing ADLs by utilizing a deep learning approach and 2) to save energy without decreasing the comfort level of the smart home inhabitant by sending recommendations about the possible energy saving actions based on the current performed activity. The overview of our framework is presented in Fig. 1.

#### B. Deep 1d-CNN

In this section, we explain in details the architecture of our proposed deep model which represents the third phase in Fig. 1.

Current common algorithms mainly use CNNs for feature extraction and classification; yet, these methods are most common in dealing with image processing. In this work, we propose a deep learning method-based CNN for daily activity recognition, which is able to handle input sequences of time series data obtained from ambient intelligent environments.

Suppose the  $i^{\text{th}}$  input time series is:

$$C_i = [c_i^{(1)} \cdot c_i^{(2)} \dots \dots c_i^{(t)} \dots \dots c_i^l]. C_i \in \mathbb{R}^d \quad (1)$$

where  $l$  is the length of the sequence,  $c_i^{(t)}$  denotes the value of time step  $t$ , and  $d$  represents the input dimension (i.e., the number of input variables).

The main component of the proposed model is the one-dimensional convolutional (1dCnov) layers. Their purpose is to automatically extract deep features from the multivariable time series input. We built our model by stacking multiple convolutional layers composed of convolution operation, activation function, max-pooling, dropout and batch normalization layers. At first, the convolution operations are applied on the sequence input to obtain the feature mapping, as formalized in the equation:

$$f^j = \sigma(b^i + \sum_i k^{ij} * x^i) \quad (2)$$

Then, the ReLU activation function is employed to insure the non-linear behavior of the network. Max-pooling layers are then used to subsample the extracted feature mapping, which

helps eliminate non-maximal values and extract the local dependency characteristics within different regions in the input sequences. After the max-pooling layer, we apply dropout which is a regularization method proposed in [25]. It is mainly used to solve the overfitting problem by randomly selecting a set of activations to be zero at training time. After the dropout, a batch normalization (BN) [26] layer is then added to accelerate the training process and improve the efficiency of our model. All the extracted features from the previous layers are then flattened into a vector to feed the input of the softmax layer (a dense layer with a softmax activation function) in order to generate the classification results. By stacking several convolution operators with the max-pooling technique, the proposed model has the ability to extract the deep features from the input sequence to create a hierarchy of abstracted features that are more robust to noise in order to successfully classify each activity. Both batch normalization and dropout are applied to each convolutional layer to improve the overall performance. Our proposed model is defined according to the network structure presented in Fig. 2.

### C. Recommendation System

In this work, we follow the passive system, which seems to be non-intrusive and more satisfying for the user. The main functionality of our system is to promote the saving of energy in smart homes by generating energy saving recommendations based on the user's current performed activity.

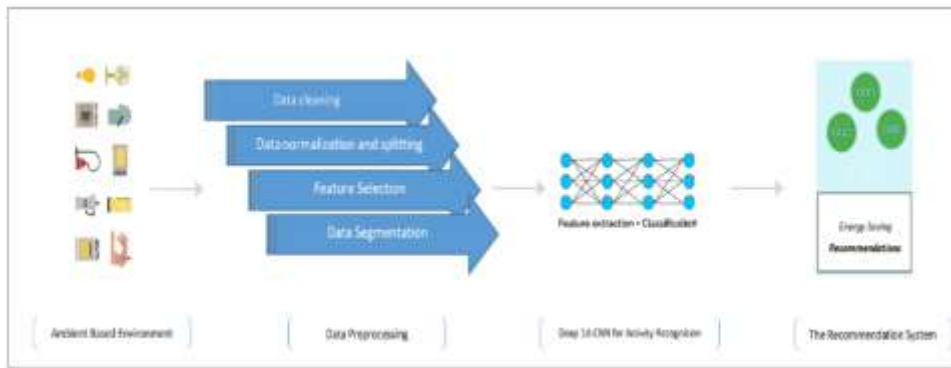


Fig. 1. Overview of our Proposed Framework.

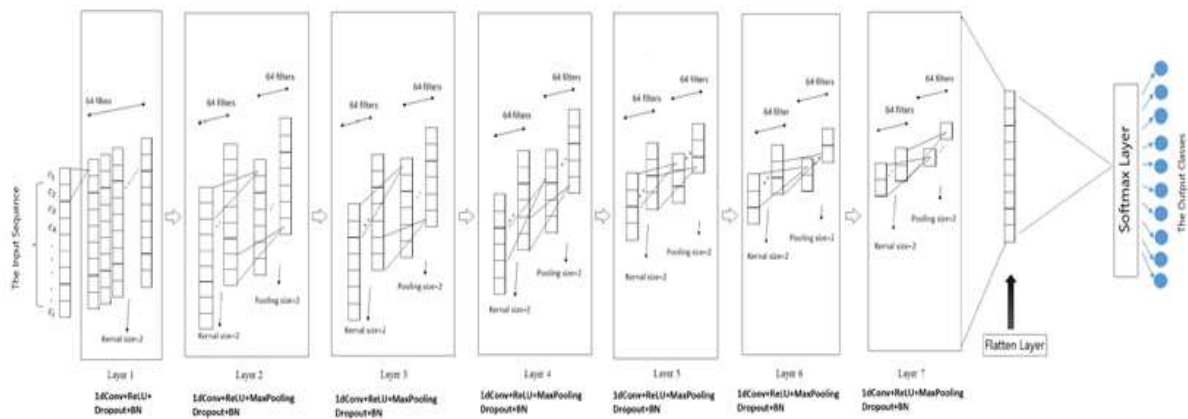


Fig. 2. The Architecture of Deep 1d-CNN for Recognizing ADLs.

At first, we designed our recommendation system to be a rule-based recommendation system. The initial strategy of our recommendation system was to recommend that the user turn off devices that are not needed in support of the current performed activity. We realized this functionality by finding the relationship between each energy-related sensor and the activities and then defining rules to perform a check for conditions in order to generate the energy saving recommendations. The goal of our system is to identify all of the energy sensors that are being used in the current performed activity. Then, for each one, the system must check whether the relationship between that sensor and the current performed activity is exciting or not. If not, energy saving recommendations will be generated to inform the user that this device, which is not needed for the current performed activity, is wasting energy.

#### IV. IMPLEMENTATION

In this section, we first introduce the dataset we used in our work and the preprocessing steps we implemented to make it suitable for further analyses. Then, we present the implementation details of our deep learning model for classifying time series data to detect ADLs in an ambient environment. Finally, we describe how we implemented our recommendation system for promoting energy saving in smart homes through recognizing ADLs.

##### A. ContextAct@A4H Dataset

We evaluated our Deep 1d-CNN model on the new real-world dataset, ContextAct@A4H [11]. It contains a large number of sensors that capture many properties related to each activity, measuring not only object interactions but also context features to improve the potential for recognizing ADLs. Although this dataset had been collected for activity recognition purposes, the dataset features many energy-related variables so it can be useful for studying energy saving solutions in smart homes (which is one of our objectives). ContextAct@A4H is a publicly available rich dataset collected using ambient sensors that reflect the context of the environment, as well as annotated ADLs. It is a real-life dataset that was gathered at the Amiquil4Home apartment using 219 of sensors to improve the potential for using the data. Ambient sensors were used to allow the observation of object usage and context conditions in all rooms and the exterior. The following context variables were measured in each room: temperature, carbon dioxide (CO<sub>2</sub>) level, noise, humidity, presence, and music information. Weather information was measured for the exterior. Appliance and object usage were measured through electric/water consumption sensors, contact sensors, and state change sensors. Other sensors obtained indirect measures of object usage (e.g., the pressure sensor in the bed). The dataset consisted of one week of captured data during the summer and three weeks in the fall. It contained 10 daily activities, which were annotated by self-reporting in place at the moment of starting and ending each activity. The activities, and their annotated names are listed in Table II.

In Fig. 3, we present an example of activity Travail (working) annotated at the time when it started and the time when it ended along with the information corresponding to the sensor events during this activity.

TABLE II. CONTEXTACT@A4H DATASET

Activity	Annotated as ...	Labelled as ...
Working	Travail	0
Cooking	Cuisine	1
Relaxing	Loisir	2
Using the Toilet	Toilettes	3
Watching TV	TV	4
Washing Dishes	Vaisselle	5
Leaving Home	Sortir	6
Taking a Shower	Douche	7
Eating	Manger	8
Sleeping	Dormir	9

```
"2016-07-30 10:18:28";"current_activity";"Travail_Start"  
"2016-07-30 10:18:29";"Presence_Bureau";"OFF"  
"2016-07-30 10:18:29";"Noise_bureau";"10.73"  
"2016-07-30 10:18:29";"Energie_Totale_Lave_Linge";"424"  
"2016-07-30 10:18:29";"Energie_Partielle_Lave_Linge";"424"  
"2016-07-30 10:18:32";"Status_TGBT";"1"  
"2016-07-30 10:18:32";"I1";"1.29945278"  
"2016-07-30 10:18:32";"PF";"1.36521101"  
"2016-07-30 10:18:32";"VIN";"235.597702"  
"2016-07-30 10:18:32";"P_active";"0.194339529"  
"2016-07-30 10:18:32";"Gaz_Logement";"83363"  
"2016-07-30 10:18:32";"E_active";"1530.326"  
"2016-07-30 10:18:32";"Freq_totale";"49.9864655"  
"2016-07-30 10:18:34";"Noise_bureau";"3.3506"  
"2016-07-30 10:18:35";"Luminosite_Cuisine";"477.76"  
"2016-07-30 10:18:35";"Noise_bureau";"7.74237"  
.  
.  
.  
.  
"2016-07-30 11:16:40";"Puissance_Lave_Vaisselle";"0.0"  
"2016-07-30 11:16:43";"Noise_bureau";"2.48645"  
"2016-07-30 11:16:45";"Noise_bureau";"11.2242"  
"2016-07-30 11:16:46";"Noise_bureau";"1.93088"  
"2016-07-30 11:16:49";"Noise_bureau";"3.94009"  
"2016-07-30 11:16:52";"current_activity";"Travail_End"
```

Fig. 3. An Example of Sensor Events Crossponding to Travail Activity in ContextAct@A4h Dataset.

##### B. Data Preprocessing

ContextAct@A4H captured about 389 features, those features are measured different type of properties using ambient sensors placed at the smart home. However, some of them are not useful for recognizing ADLs (i.e., music features). Thus, in the preprocessing phase, we removed these unwanted observations from the dataset. Moreover, we converted all needed contextual features to numerical information. For example, binary sensors, such as contact sensors, were given one of two values: open (if a door opened) or closed (if it did not). Hence, we converted the measured unit from contextual information to either 1 or 0. Meanwhile, we cleaned corrupted and repeated data, which could have been caused by hardware failures or problems in the data transfer. After the aforementioned actions were completed, the total number of records was equaled to 642,001.

As with any data pre-processing, we applied data normalization, which is the process of rescaling all feature values in a dataset into a given range. Specifically, we used a min-max scaler, which changed all the features to be between 0 and 1. After data normalization, we split the data randomly into 70% for training and 30% for testing. After making the dataset suitable for further analysis, we applied the feature selection method explained in the next subsection. After making the dataset suitable for further analysis, we applied the feature selection method explained in the next subsection. After making the dataset suitable for further analysis, we applied the feature selection method explained in the next subsection.

### C. Feature Selection

In many practical applications of ML algorithms, data that has been pre-processed can result in a large number of features, and it is often necessary to minimize this number during the training phase to allow effective classification. Additionally, supervised learning models that train on data with many features might overfit the model. As we mentioned previously, the dataset contained about 389 features of sensor readings with values ranging from 0 to 1. Our purpose was to reduce the number of features and choose only those that would contribute the most to increase the overall performance of our classification model. Therefore, we applied a common feature selection technique in which we extracted a feature importance rank using different algorithms. Feature importance provides a measure that indicates the value of each feature in the construction of a model. The more an attribute is used to build a model, the greater its relative importance. In scikit-learn, decision tree models and ensembles of trees provide a `feature_importances_` attribute when fitted [27]. The `feature_importances_` attribute used to extract the relative importance of the features from the model. In this work, we compute the relative importance of each attribute in ContextAct@A4H using different algorithms such as Random Forest and AdaBoost since they are support `feature_importances_` attribute (see Fig. 4).

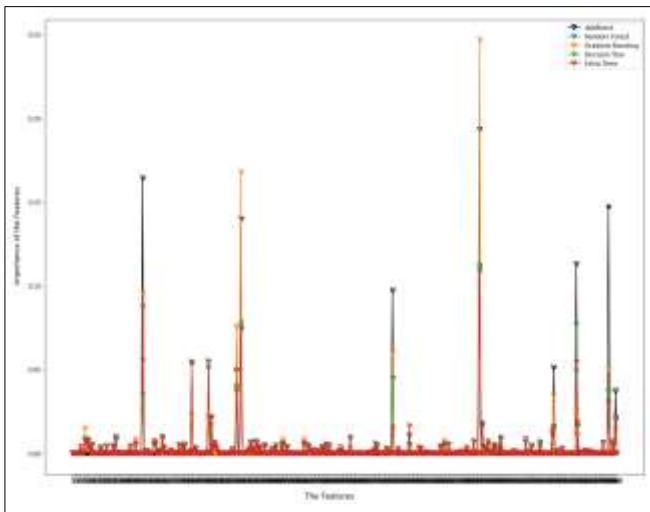


Fig. 4. Feature Importance of Entire Dataset.

After determining the importance of each feature in relation to the activities, we chose the ones with a rank higher than 0.003. Hence, we had a total of 48 features to use in our classification model. Those features are measured different type of properties using ambient sensors at the smart home.

### D. Data Segmentation

In activity recognition, data segmentation takes place after the preprocessing step and the feature selection mechanism. For an online deep learning system to effectively classify a particular feature vector representation of the sensor data, the system must include a mechanism for selecting those sensor events from an incoming stream of sensor data. This process is known as segmentation and aims to select the sensor data that lies between activity boundaries. It identifies segments that contain information about the activities to be recognized. Many segmentation methods based on the windowing of streaming data have been used in studies on activity classification, such as time- and sensor-based windowing [28]. Time based windowing breaks the dataset into equal periods of time to segment the data. This approach is favorable when using streamed data where the sensors update continuously in time as in case of our dataset [29]. Thus, we applied time-based segmentation with fixed size and overlapping windows to take the values of the sensors observed in the previous segment. This is because it is generally assumed that, due to the higher number of data points, overlapping sliding windows increase the performance of HAR classifiers compared to non-overlapping ones [30]. In our work, the feature selection is used to reduce the number of features by selecting the most importance ones, while the segmentation step is performed to divided the temporal dimension of the streaming data into windows. All these windows are shared same number of steps, which are vary in time, but they are represented by the same features (which have been selected previously).

In the experiment, we evaluated the effect of varied window sizes and different overlapping sizes on the recognition results of the proposed system. We got the best results when we set the time period of the window to be 80 time steps with overlapping started after a five-step distance while each time step represented by the 48 features. This process yielded 89,971 and 38,398 samples for the training and testing sets, respectively. Each window usually contained time steps associated with different labels. We chose for majority labeling, where the label of each window was specified as the most frequent label in those of the timestamps.

### E. Implementation of Deep 1d-CNN

We implemented our deep learning model with Google Colaboratory (also known as Colab), which is a cloud service based on Jupyter Notebook. We used Python runtimes pre-configured with essential libraries, such as TensorFlow and Keras, to build and train our neural networks. The model training and classification were run on an Nvidia Tesla K80 GPU with 2496 CUDA cores and 12 GB RAM. The neural networks were trained in a supervised way by back-propagating the gradients from the softmax layer through to the

convolutional layers. The categorical cross-entropy used was the loss function. For optimization, we used Adam [31] with a learning rate of 0.0001. The training was done on 89,971 samples, and the testing was done on 38,398 samples with a batch size equal to 1,024. To build the topology of Deep 1d-CNN, we performed many experimental studies to define the suitable hyper-parameters for our model. In the end, we decided to have 7 one dimensional convolutional layers, each with 64 kernels, a filter size of 2, and a ReLU activation function. A max-pooling of size two was included in all convolutional layers except in one layer because the temporal dimension of the feature map at the last layer became one; therefore, the max-pooling had to be avoided in any layer. To avoid overfitting, a dropout layer was used in each convolutional layers as a regularization mechanism with a rate equal to 0.2. we also implemented a batch-normalization layer to accelerate the training and improve the overall performance. At the end, all outputs from the previous layers were flattened into a one-dimensional vector, which was then connected to a softmax classifier to defined the output probabilities of the class labels. The hyper-parameters of our deep learning model for the ContextAct@A4H dataset are provided in Table III.

TABLE III. THE RELEVANT PARAMETERS OF EACH LAYER IN DEEP 1D-CNN

Num.	Components	Layer Parameters
Layer 1	1dConv	64*2
	Activation	ReLU
	Dropout	Rate=0.2
	Batch Normalization	-
Layer 2 to Layer 7	1dConv	64*2
	Activation	ReLU
	MaxPool	2
	Dropout	Rate=0.2
	Batch Normalization	-
Layer 8	Flatten Layer	-
Layer 9	Dence Layer	Softmax

#### F. Implementation of the Recommendation System

Based on the current performed activity predicted by the deep model, our system detected which devices were turned on but not needed for that activity. We realized this functionality by applying rules that checked whether the relation between the current performed activity and the devices that were turned ON existed in the predefined relationships. In another words, our recommendation system used the rules and the relationships to identify appliances that were wasting energy while a certain activity was being performed by the user. To implement this method, we followed several steps:

- First, we extracted the energy sensors from the dataset (all from type switch and electrical outlet). They gave two readings: 0 when the attached device was turned OFF and 1 when it was ON.
- Then, we determined where each device was located.

- After that, we identified the relationship between the sensors and the activities.
- Finally, we defined rules to perform a check for conditions.

Table IV provides examples of what we got after implementing these steps.

Consider the following example, if we assume that our Deep 1d-CNN model predicted the current performed activity is watching TV, the recommendation system would check the value of each energy sensor at the last time period of that activity. If the value of PC10 was ON, then the system would recommend the user to turn it OFF since there are no relationship between this sensor and the watching TV activity.

TABLE IV. IMPLEMENTATION OF THE RECOMMENDATION SYSTEM

Energy-related Sensor ID in the Dataset	Sensor Location	Relation to Activity
PC10 (Smart electrical outlet)	Bedroom	Sleeping
tv_salon_real_status (Switch )	Living Room	Watching TV, Relaxing, and Eating
<b>Defined Rule</b>	If the value of the energy sensor was not equal to 0 and the relationship between this sensor and the current performed activity was false, then a recommendation must send to the user to turn OFF the unattended device.	

#### V. PERFORMANCE EVALUATION AND DISCUSSION

The results of the performance of our model are presented in this section. We also investigated the performance of our model in case if we removed the max pooling layers in order to see its effectiveness in our network. In addition, to show the advantage of our proposed model, we compared our network with baseline results obtained from deep models. Finally, we show the results of the recommendation system in terms of how much energy savings could be gained through recognizing ADLs.

##### A. Performance Evaluation of Deep 1d-CNN

1) *Performance measurement:* Classification accuracy is not an appropriate measure of performance since activity datasets collected in natural scenes are often imbalanced between classes. In ContextAct@A4H, some classes (e.g., sleeping and leaving home) contain more number of examples while other classes have less. Therefore, we evaluated the models using other metrics that were suitable for evaluating class-imbalanced problems. Those metrics—precision, recall, and F1 score—have also been used in previous works such as in [16].

Precision determines the ratio of the relevant points chosen by the model to the total selected points, as shown in the following equation, where TP is the number of true positives, and FP is the number of false positives:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$



Recall is the ratio of relevant points selected by the model to the overall total of relevant points, as shown in the following equation, where TP is the number of true positives, and FN is the number of false negatives:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

The F1 score is defined in terms of both precision and recall as follows:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

2) *Experimental results:* The classification results of the proposed method on the ContextAct@A4H dataset are presented in this subsection. We calculated the previously described metrics, namely, precision, recall, and F1 score. We also plotted the accuracy and loss curves during the training and testing process for Deep 1d-CNN.

As shown in Table V, the precision, recall, and F1 score results indicate that applying 1d-CNNs for the time series classification was effective due to their ability to extract very informative deep features from each sequence input to detect the ADLs.

Meanwhile, according to Fig. 5, the training and testing loss of our model kept decreasing until the last epoch, in which both remained in the same range with a slight decrease in training loss. When we stopped the training process, the training loss equaled to 0.21 and validation loss was 0.28. Moreover, the testing accuracy improved until it reached 0.90, while the training accuracy was 0.92. Both then stayed in their same range with no further increases or decreases where we stopped the training process.

The two curves demonstrate that our model did not suffer from an overfitting problem due to add dropout layers in each convolutional layer, which provide regularization and, hence, prevent overfitting. To support our observation, we trained the model without the dropout layers to see how the training process would progress. We observed that the model without the dropout layers started overfitting the data from the beginning, and we had to stop the training at Epoch 30 where the loss function values for the training and testing data were 0.02 and 0.82, respectively. In addition, the gap between training and testing accuracy got bigger, which also indicates that the model overfitted the data. Finally, the results of training our model with BN provided a lower loss function value than the model without BN for both the training and testing data and also resulted in a higher model accuracy rate.

TABLE V. THE CLASSIFICATION RESULTS OF OUR PROPOSED MODEL

Results of Deep 1d-CNN	
Precision	0.91
Recall	0.89
F1 score	0.90
Time Per Epoch	4 sec

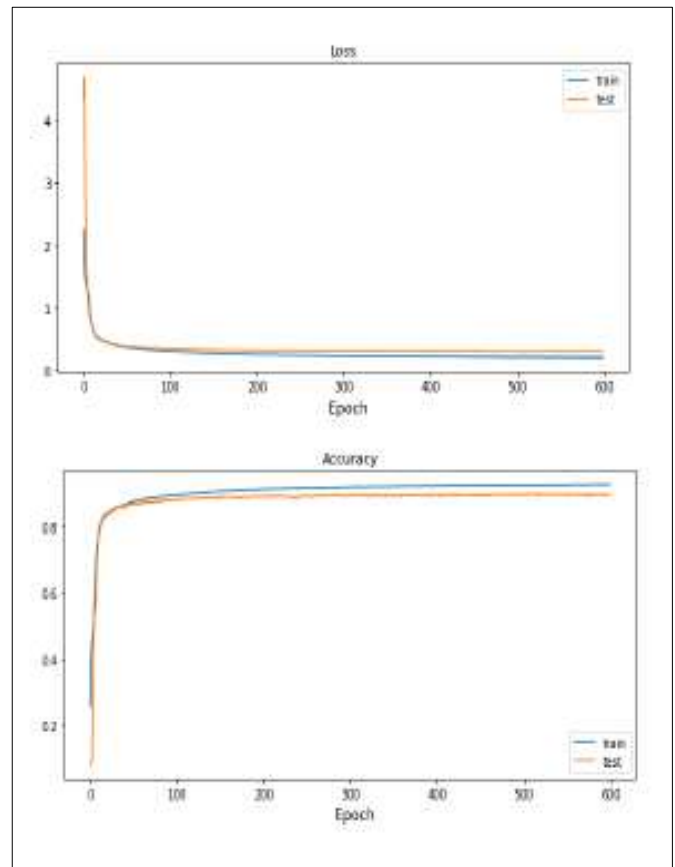


Fig. 5. The Loss and Accuracy Curves of Our Model.

3) *Comparative Results on Deep 1d-CNN with Other Proposed Deep Models:* In this section, we describe the results of our model obtained on the ContextAct@A4H dataset compared to other deep learning approaches proposed for recognizing ADLs using the aforementioned metrics. Since, as said previously, ContextAct@A4H is a newly collected dataset, it has been used in few studies for recognizing ADLs [32,33]. However, non-of them used it along with deep model to solve the complexity of activity recognition, thus, we could not compare our work with them. To overcome such limitation, we chose two deep models proposed in other works for recognizing ADLs and we implemented in our dataset, ContextAct@A4H, to illustrate the advantages of our model Deep 1d-CNN. These two models are 1D-CNN [15] and LSTM [17] and the result of the comparison is presented in Table VI. We train the 1D-CNN model with learning rate equal to 0.0001 while the LSTM model was train with learning rate equal to 0.00001. In both model, we had to stop the training from early epoch (30 epochs for 1D-CNN and for LSTM, we stopped it at epoch 10) since they both started to overfit the training data. In the LSTM model, there was no regularization method used to reduce the overfitting problem, while in the 1D-CNN model, they applied one dropout layer, however, it implemented after the fully connected layer not in the convolutional layers.

TABLE VI. PERFORMANCE COMPARISON OF DEEP 1D-CNN WITH OTHER DEEP MODELS

Model	Precision	Recall	F1
1D-CNN [15]	0.84	0.84	0.84
LSTM [17]	0.81	0.79	0.80
Deep 1d-CNN (Our Model)	0.91	0.89	0.90

### B. Energy Savings through Recognizing ADLs

In this section, we evaluate the potential energy savings that could be achieved by our recommendation system in the smart home through recognizing ADLs. We followed an evaluation approach in which we calculated the amount of energy consumed over a period of time with and without the application of our system to find the total percentage of energy savings. As we have mentioned before, the dataset we used consisted of a sequential set of fixed length windows. In this evaluation, we were interested in calculating the energy consumed during each activity by utilizing the last time step of the window before and after applying the system. In this experiment, we followed the assumption that the user was performing one activity at a time.

First, we mathematically calculated the number of energy sensors that were on in the last time period for each predicted activity without applying the energy saving approach. Then, we estimated the total energy consumption per hour using the following equation:

$$totalEC = (TotalEnergyON * time) / 3600 \quad (6)$$

where *totalEC* is the amount of energy consumed before applying the energy saving approach, *TotalEnergyON* is the number of energy sensors that were ON before the energy saving recommendations, and *time* in our case, is the number of samples, since those samples were recorded sequentially over a period of time.

Second, we calculated the number of energy sensors that remained ON in the same time period after applying the energy saving recommendation. Then, we estimated the total energy consumption per hour using the following equation:

$$savingEC = (RemainingEnergyON * time) / 3600 \quad (7)$$

Where *savingEC* is the amount of energy consumed after applying our energy saving system, *RemainingEnergyON* is the number of energy sensors that are kept ON after the energy saving recommendations, and *time* is equal to the number of the samples. Once we determined the energy consumption with and without the energy saving system, we calculated the percentage of energy that have been saved by our system using the following equation:

$$TotalES = totalEC - savingEC / totalEC * 100 \quad (8)$$

Where *totalEC* is the total energy consumption without applying the energy saving system, and *totalEC* represents the amount of consumed energy after applying the system.

As seen in Fig. 6, a significant proportion of energy could be saved by utilizing the recognition of ADLs for an energy saving purpose.

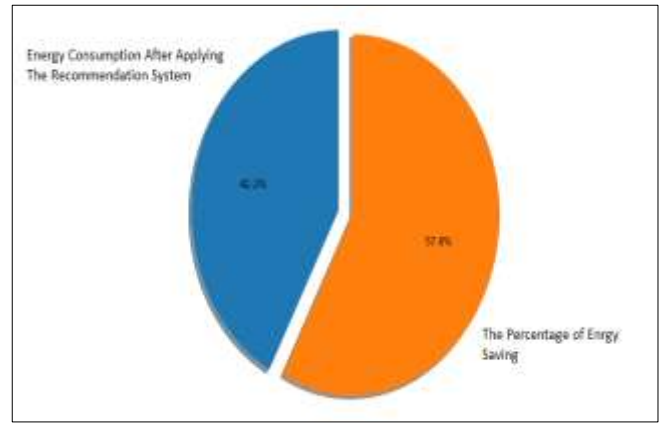


Fig. 6. The Proportion of Energy Saving through Recognizing ADLs.

## VI. CONCLUSION

In this work, deep one-dimensional convolutional layers equipped with the max-pooling technique were proposed to deal with time series data to detect ADLs in ambient intelligent environments. Our Deep 1d-CNN unifies feature extraction and classification; it can automatically discover and extract the internal structure of the input time series data and learn deep features to classify ADLs. We evaluated our proposed method on a new real-world dataset, ContextAct@A4H. The results demonstrated that 1d-CNN is one of the most competitive candidates for time series classification owing to its ability to learn more robust deep features. We also expanded our experiment to evaluate the potential energy saving in the smart home by utilizing the recognized activity of our deep model to detect which devices were wasting energy and encourage the user to perform optimization actions. The experiment indicated that recognizing ADLs can achieve an energy savings of around 50%.

The framework proposed in this work has some limitations. They can be seen as extensions and can be addressed in the future. Regarding the proposed deep model, our Deep 1d-CNN achieved good performance in recognizing ADLs, however, we designed our model to only learn single-user activities. Another limitation of our deep model is that most of the parameters were determined through lots of experiments, which is time consuming; so, designing a method of hyper-parameter optimization is ongoing. On the subject of the energy saving recommendation system, the drawback of this approach is that the user might have purposely kept the devices that were detected as wasting energy turned on. However, in using this evaluation approach, we only attempted to show the maximum potential energy savings and demonstrate how recognizing ADLs can have a great impact on energy saving in smart homes.

Regarding directions for future work, we will investigate the problem of recognizing multi-user activities. Moreover, we intend to explore the problem of data imbalance, which is a common problem for real-life HAR. For the recommendation system, we plan to allow the users to send feedback to the system so that it can learn from this feedback in order to enhance its services.

REFERENCES

- [1] Jiang, J., Pozza, R., Gunnarsdóttir, K., Gilbert, N., & Moessner, K. (2017). Using sensors to study home activities. *Journal of Sensor and Actuator Networks*, 6(4), 32.
- [2] Guo, X., Shen, Z., Zhang, Y., & Wu, T. (2019). Review on the application of artificial intelligence in smart homes. *Smart Cities*, 2(3), 402–420.
- [3] Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119, 3–11.
- [4] Richardson, F., Reynolds, D., & Dehak, N. (2015). Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, 22(10), 1671–1675.
- [5] Chan, T. H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2015). PCANet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12), 5017–5032.
- [6] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *International Conference on Engineering and Technology (ICET)* (pp. 1–6). IEEE.
- [7] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
- [8] Oukrich, N. (2019). Daily HAR in smart home based on feature selection, neural network and load signature of appliances (Doctoral dissertation).
- [9] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery*, 33(4), 917–963.
- [10] Lima, W. S., Souto, E., Rocha, T., Pazzi, R. W., & Pramudianto, F. (2015, July). User activity recognition for energy saving in smart home environment. In *2015 IEEE Symposium on Computers and Communication (ISCC)* (pp. 751–757). IEEE.
- [11] Lago, P., Lang, F., Roncancio, C., Jiménez-Guarín, C., Mateescu, R., & Bonnefond, N. (2017, June). The ContextAct@ A4H real-life dataset of daily-living activities. In *International and Interdisciplinary Conference on Modeling and Using Context* (pp. 175–188). Springer, Cham.
- [12] Ramasamy Ramamurthy, S., & Roy, N. (2018). Recent trends in machine learning for human activity recognition—A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), e1254.
- [13] Nweke, H. F., Teh, Y. W., Al-Garadi, M. A., & Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, 105, 233–261.
- [14] Sadouk, L. (2018). CNN approaches for time series classification. In *Time series analysis-data, methods, and applications*. IntechOpen.
- [15] Singh, D., Merdivan, E., Hanke, S., Kropf, J., Geist, M., & Holzinger, A. (2017). Convolutional and recurrent neural networks for activity recognition in smart environment. In *Towards integrative machine learning and knowledge extraction* (pp. 194–205). Springer, Cham.
- [16] SEDKY, M., HOWARD, C., Alshammari, T., & Alshammari, N. (2018). Evaluating machine learning techniques for activity classification in smart home environments. *International Journal of Information Systems and Computer Sciences*, 12(2), 48–54.
- [17] Singh, D., Merdivan, E., Psychoula, I., Kropf, J., Hanke, S., Geist, M., & Holzinger, A. (2017, August). Human activity recognition using recurrent neural networks. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (pp. 267–274). Springer, Cham.
- [18] Al Machot, F., Ranasinghe, S., Plattner, J., & Jnoub, N. (2018, March). Human activity recognition based on real life scenarios. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom) Workshops* (pp. 3–8). IEEE.
- [19] Arifoglu, D., & Bouchachia, A. (2017). Activity recognition and abnormal behaviour detection with recurrent neural networks. *Procedia Computer Science*, 110, 86–93.
- [20] Nguyen, T. A., & Aiello, M. (2013). Energy intelligent buildings based on user activity: A survey. *Energy and Buildings*, 56, 244–257.
- [21] Alhamoud, A., Ruettiger, F., Reinhardt, A., Englert, F., Burgstahler, D., Böhnstedt, D., & Steinmetz, R. (2014, September). Smartenergy. kom: An intelligent system for energy saving in smart home. In *39th Annual IEEE Conference on Local Computer Networks Workshops* (pp. 685–692). IEEE.
- [22] Thomas, B., & Cook, D. (2016). Activity-aware energy-efficient automation of smart buildings. *Energies*, 9(8), 624.
- [23] Mubdir, B., Al-Hindawi, A., & Hadi, N. (2016). Design of smart home energy management system for saving energy. *European Scientific Journal (ESJ)*, 12(33), 521. *Scientific Journal (ESJ)*, 12(33), 521.
- [24] Cottone, P., Gaglio, S., Re, G. L., & Ortolani, M. (2015). User activity recognition for energy saving in smart homes. *Pervasive and Mobile Computing*, 16, 156–170.
- [25] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.
- [26] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- [27] Brownlee, Jason . (2014, July 14). Feature Selection in Python with Scikit-Learn. Retrieved April 22, 2020, from <https://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn/>.
- [28] Yala, N., Fergani, B., & Fleury, A. (2015, September). Feature extraction for human activity recognition on streaming data. In *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)* (pp. 1–6). IEEE.
- [29] Quigley, B., Donnelly, M., Moore, G., & Galway, L. (2018). A Comparative Analysis of Windowing Approaches in Dense Sensing Environments. In *Multidisciplinary Digital Publishing Institute Proceedings (Vol. 2, No. 19, p. 1245)*.
- [30] Dehghani, A., Sarbishei, O., Glatard, T., & Shihab, E. (2019). A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors. *Sensors*, 19(22), 5026.
- [31] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [32] Molano-Pulido, J., & Jiménez-Guarín, C. (2018). SEABIRD: Adaptable Daily Living Activity Identification from Sensor Data Streams. *Procedia Computer Science*, 130, 939–946.
- [33] Lago, P., Roncancio, C., Jiménez-Guarín, C., & Labbe, C. (2017, August). Representing and learning human behavior patterns with contextual variability. In *International Conference on Database and Expert Systems Applications* (pp. 305–313). Springer, Cham.