Cluster Computing (2022) 25:2085–2105
https://doi.org/10.1007/s10586-021-03350-z

Artificial intelligent system for multimedia services in smart home environments

Albert Rego[1] · Pedro Luis González Ramírez[1] · Jose M. Jimenez[1] · Jaime Lloret[1]

Received: 28 March 2021 / Revised: 1 June 2021 / Accepted: 19 June 2021 / Published online: 6 July 2021

Abstract

Internet of Things (IoT) has introduced new applications and environments. Smart Home provides new ways of communication and service consumption. In addition, Artificial Intelligence (AI) and deep learning have improved different services and tasks by automatizing them. In this field, reinforcement learning (RL) provides an unsupervised way to learn from the environment. In this paper, a new intelligent system based on RL and deep learning is proposed for Smart Home environments to guarantee good levels of QoE, focused on multimedia services. This system is aimed to reduce the impact on user experience when the classifying system achieves a low accuracy. The experiments performed show that the deep learning model proposed achieves better accuracy than the KNN algorithm and that the RL system increases the QoE of the user up to 3.8 on a scale of 10.

Keywords Deep learning · Classification · Smart home · Reinforcement learning · Multimedia

1 Introduction

Integrated networks are communications networks that use devices from multiple manufacturers, protocols, and service operators. Currently, using integrated networks, it is possible to interact with sensors and everyday devices, deployed in what is known as the Internet of Things. IoT is a concept that is being applied in multiple areas, among which we can highlight Smart Home, Smart Cities, Smart Agriculture, Smart Grid, Smart Health, and Wearables.

One of the areas in which the application of the IoT is increasing day by day is in Smart Home. According to [1], in 2015 the smart home sector had a global value of $ 47 billion and they forecast a growth of 14% per year in the period 2016 to 2022. According to data from the observatory [2], the Italian market for the smart home sector in 2018 had purchase estimates of 380 million euros, which represents an increase of 52% compared to 2017. In addition, it indicated data from other European countries such as Germany (1.8 billion euros, an increase of 39%), the United Kingdom (€ 1.7 billion, 39% increase), France (€ 800 million, 47% increase) and Spain (€ 300 million, 59% increase).

We can define Smart Home as a system that allows the monitoring and control of a home or office easily. Its application is carried out by both companies and individuals. In order to use it, we must have access, through an integrated network, to the devices located in the home or office. Its implementation can be very broad and located on very different devices. It can go from simple monitoring with cameras for security control, to something as complex as the use of smart refrigerators, which can place orders automatically, when they detect the lack of food. Its most common applications are the control of lighting, gas, water, air conditioning, doors and windows, security cameras, weather sensors, irrigation the home theater, etc. Some of these applications are multimedia applications. In multimedia services, the Quality of Experience (QoE) is the

✉ Jaime Lloret
jlloret@dcom.upv.es

Albert Rego
alremae@teleco.upv.es

Pedro Luis González Ramírez
pedgonr4@doctor.upv.es

Jose M. Jimenez
jojiher@dcom.upv.es

[1] Instituto de Investigación para la Gestion Integrada de zonas Costeras, Universitat Politècnica de València, Carrer del Paranimf, 1, 46730 Grau de Gandia, València, Spain

most important metric, since it provides a measure of how good a service is from the user's point of view [3].

If we observe carefully, we will realize that the use of electrical energy is necessary for most of the activities that we carry out in our homes throughout the day. We must be responsible with the use of energy due both to the depletion of certain energy sources, and to the impact that occurs on the environment. On a regular basis, at home, we have a large number of appliances such as a refrigerator, freezer, dishwasher, washing machine, dryer, oven, stove, etc. that we use even several times a day. These appliances can often account for half of the energy consumption in the home. Also, it is important to control the consumption of light and the appliances used for air conditioning. In addition, the consumption of electronic devices, in general, is usually higher than it should be. In general, our televisions, multimedia players, etc have a system called standby that allows them to never be unplugged from the current, so their electricity consumption is permanent.

Artificial Intelligence can be used to improve and optimize energy consumption in our homes. One of the biggest keys to using AI is its learning phase. The key to optimizing our energy consumption at home is data. As a smart home is an automated environment, we can monitor and get to capture the patterns of daily activities, which are provided by sensors through information technologies. The greater the amount of information available about our habits, the greater the adjustment in energy consumption savings can become. Using learning, automatic or supervised, we can make the control devices capable of self-programming. In addition, we can apply Deep Learning, which allows us to use logical structures very similar to those of organization of the nervous system of a mammal. By using Deep Learning on the control devices that we have at home, we will be able to optimize energy consumption in our smart home.

In this paper, we present an intelligent system in Smart Home environments that manages the set of nodes and services of the environment, enabling or disabling them based on the predictions of user's service consumption. This system is aimed to improve the results of a deep learning classification system when the algorithm is still learning from the user. Deep Learning works better with a huge number of data and our proposal, based on RL is aimed to provide better flexibility to the system.

The rest of this document is structured as follows. Section 2 presents some of the most relevant works related to our study. The network and system environment and architecture are described in Sect. 3. Section 4 briefly defines the data preprocessing and the classification model, based on deep learning. Section 5 describes the RL algorithm and all its components. Section 6 presents the evaluations of the system. Finally, Sect. 7 concludes the paper and presents some future works.

2 Related work

There is a large number of articles in which the authors have studied smart homes.

Some authors like [4–8] present studies on IoT and Smart Home. Vasicek et al. [4] present the IoT concept in a Smart Home. They use IoT devices to create a smart home, without the need to rebuild the home. Jie et al. [5], present a highly scalable architecture, applying IoT technologies, where they can integrate many applications using a uniform interface, to develop a smart home. Khan et al. [6] present the design of an IoT Smart Home System (IoTSHS), which was designed, programmed, manufactured and tested with excellent results. The authors stress that it can benefit all parts of society by providing advanced remote control for the smart home. Malche and Maheshwary [7] analyze the functions of a smart home and its applications and introduce an architecture that they call FLIP (Frugal Labs IoT Platform), to build smart homes enabled for IoT. Yang et al. [8] present a study examining service characteristics of smart homes in 216 samples in Korea. They also do a study of personal characteristics in the behavior of users.

There are authors like [9–13] who present reviews about IoT and Smart homes. Risteska Stojkoska and Trivodaliev [9] propose a holistic framework, which incorporates different components from IoT architectures/frameworks proposed in the literature. In addition, they identify a management model for the proposed framework, identifying the tasks to be carried out. Alaa et al. [10] present a review study, where they formalize a taxonomy focused on three areas, (1) smart homes, (2) apps, and (3) IoT in three major databases, namely, Web of Science, ScienceDirect, and IEEE Explore. Kuzlu et al. [11] present a review where they compare wireless and wired communication technologies in local area networks applied to Smart homes, taking into account the standards, protocols, data rate, coverage, and adaptation rate. Kamel and Memari [12] present a review, whose main objective is to classify the different types of Smart homes into three main groups, from the energy point of view. Groups are established at homes with energy monitoring systems, systems with control capabilities, and systems with advanced data processing capabilities. Finally, Jia Chen et al. review, in [13], the use of IoT in*home systems and applications for health monitoring. They reviewed the key factors that caused the growth of IoT monitoring at home. Then, they presented the lastest advances of the architecture of these systems.

Lastly, they discussed future outlooks and personal recommendations.

Other authors [14–18] present studies from the point of view of access security to IoT devices in smart homes. Apthorpe et al. [14] present the study of four Smart home IoT devices. They find that network traffic rates can show user interactions, even when the traffic is encrypted. For this reason, they indicate the need for technological solutions to protect the privacy of users. Augusto-Gonzalez et al. [15] present the GHOST framework (Safe-Guarding Home IoT Environments with Personalized Real-time Risk Control), which aims to provide cyber security to residents of smart homes. They do it through a new reference architecture, for smart home security and its users. Lin and Bergmann [16] establish key requirements to give reliability in smart homes in the future. They propose a gateway architecture, to have a high availability of the system and devices with limited resources. Meng et al. [17] present the most popular architecture for smart home platforms, detailing the functions of each of its components. They also comment on the main security and privacy challenges of the platforms and review the state of the art of the proposed countermeasures. Ammi et al. [18] propose a novel Blockchain-based solution for secure smart home systems, using a combined hyperledger fabric and hyperledger composer. Another important aspect of the proposed solution is the mapping of the attributes of a smart home to those from the hyperledger composer. This mapping allows for a customized, designed-for-purpose solution that can meet the security requirements for IoT-based smart homes.

Atat et al. [19] present the cyber-physical systems (CPS) taxonomy by providing a broad overview of data collection, storage, access, processing, and analysis. These systems are foreseen to revolutionize our world via creating new services and applications in a variety of sectors, such as environmental monitoring, mobile-health systems, intelligent transportation systems, and so on. This is the first panoramic survey on big data for CPS, where their objective is to provide a panoramic summary of different CPS aspects. Also, they provide an overview of the different security solutions proposed for CPS big data storage, access, and analytics. In addition, they discuss big data meeting green challenges in the contexts of CPS.

There are authors [20–23] who study energy management in the Smart home environment. Collotta and Pau [20] propose an energy management system for smart homes, using Bluetooth Low Energy (BLE) [21] for communications, together with a home energy management (HEM) scheme. Their results show the efficiency of the proposed system, since they reduce the peak load demand, and the charges for electricity consumption, thus increasing the comfort of its users. Al-Ali et al. [22] present an Energy Management System (EMS) for smart homes. They use Business Intelligence (BI) and Big Data analytics software to manage energy consumption, satisfying consumer demand. Xia et al. [23] propose an edge-based energy management framework, which provides a low cost of electricity and saves on infrastructure construction. They have implemented a prototype, the results of which show a reduction in the cost of electricity of 82.3%, compared to similar cases. Celik et al [24] present a review of the antecedents in modeling of residential load, demand-side management (DSM) and demand response (DR), in the settings of a home and in a neighborhood area. The objective is to classify the structure and coordination techniques of energy management, from previous research. Authors as Wu et al. [25] discover the relations between the trend of the big data era, and that of the new generation green revolution, through a comprehensive and panoramic literature survey in big data technologies toward various green objectives and a discussion on relevant challenges and future directions.

Amjad et al. [26] propose a cognitive edge-computing-based framework solution, to integrate the advancement of edge computing resource requirement schemes as well as the resource allocation schemes found in the literature for enterprise cloud; to attain a universal resource allocation framework for IoT. Others Authors as Jararweh et al. [27] present a novel experimental framework for IoT-based environmental monitoring applications, using concepts from Data Fusion (DF) and software defined systems (SDS). It is built on top of the software defined networking platform where the core components (the host, switch and the controller) are expanded to support other software defined systems components (such as software defined storage and security) and enable the applications of different DF techniques in IoT environments.

Studies [28–32] related to the application of AI in the field of Smart Home can also be cited. Sodhro et al. [28] state that the convergence of IoT and AI promotes energy-efficient communication in smart homes. Its main objective is to optimize the Quality of Service (QoS) of video transmission, which is carried out using wireless micro medical devices (WMMD), in smart healthcare homes. Guo et al. [29] make reviews of the literature and existing products to define the functions and roles of AI in Smart homes. They point out the existence of a delay between the literature and the products. Sepasgozar et al. [30] reviewed the applications of the IoT in homes, to make them intelligent, automated and digitized in many of its aspects. They have studied the literature on the use of IoT, AI and geographic information systems (GIS) in smart homes. They state that there is a considerable gap in the integration of AI and IoT and the use of geospatial data, in the field of Smart Home. Song et al. [31] present frameworks of centralized and distributed AI-enabled IoT networks. Key technical challenges are analyzed for different network architectures.

Deep reinforcement learning (DRL)-based strategies are introduced and neural networks-based approaches are utilized to efficiently realize the DRL strategies for system procedures. Different types of neural networks that could be used in IoT networks to conduct DRL are also discussed. Lloret et al. [32] proposed an intelligent system for detecting elderly problems and assist them. They proposed a communication architecture and designed a software application.

Several works have been carried out about Smart Home from different points of view. However, the role that an automated intelligent management system can play in Smart Home environments needs to be discussed. Consequently, we propose a new role for AI in this scenario. Our proposed system works along with the user to reduce the intrusion that an automatized service management can introduce to the user's experience. Moreover, the system is oriented to especially reduce the impact of bad predictions on multimedia services, trying to guarantee a good QoE from the user's point of view. In order to overcome the possible difficulties that a deep learning classifying method may provide to the system, we introduce the use of an RL-adapted method into the Smart Home. RL has been used in other works like in [33] due to its performance.

3 Proposed architecture

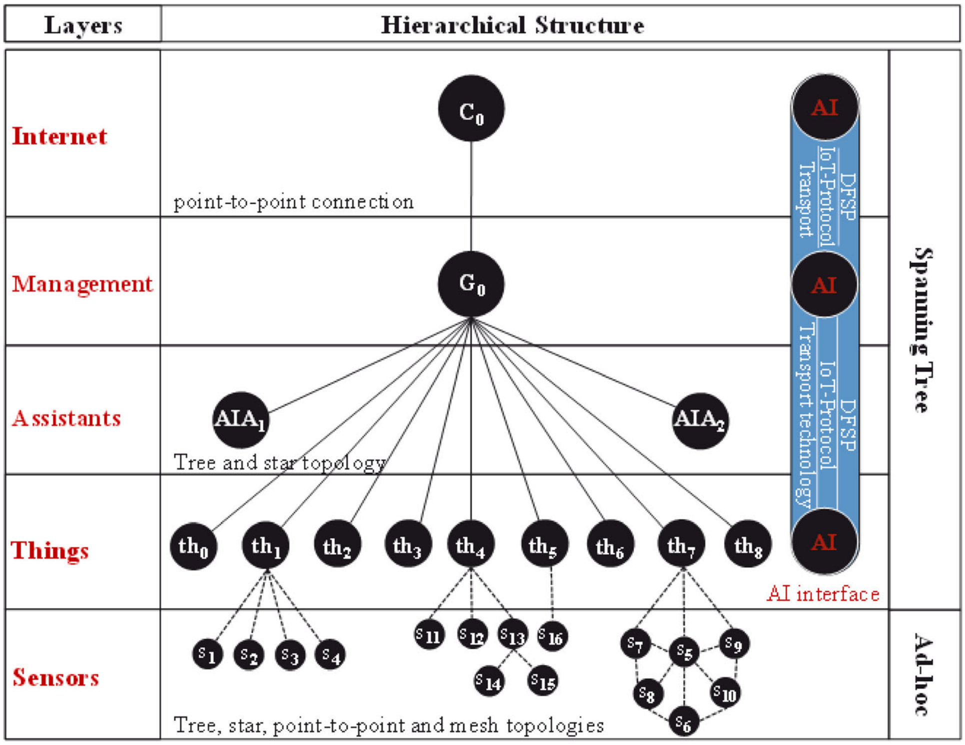
In this section, the architecture of the proposal is detailed. First, the network architecture is explained. Then, the intelligent system is detailed, explaining the function of each module that composes it.

3.1 Network architecture

The architecture of this proposal is an architecture with centralized management based on AI [34, 35]. This architecture is divided into five logical layers to organize and maintain divided the functions of each object connected to the network. The management layer (Layer 4) centralizes the information and stores the data of the features of the connected objects (dataset) and the parameters and statistics of the network. The AI uses this information to create workgroups (grouping) and routing, and other functions that keep the network operational. The multiprotocol IoT Gateway is the device in this layer in charge of doing this work and has the capacity to handle different interconnection technologies, store information, host an AI, process information and control internet access. In the internet layer (layer 5), the AI within the cloud can choose the IoT Platform according to the type of parameter and the capacity to process large volumes of data. The rest of the layers are located below as Artificial Intelligence Assistants (AIA) in layer 3, smart things (th) in layer 2 and smart sensors and actuators in layer 1. In this way, the functionalities are separated and kept as a stand-alone system. Fig. 1 shows the hierarchical distribution of the architecture with the objects represented as nodes.

This architecture is used to design Smart IoT-Networks, which contain interconnected objects with integrated AI (Smart Things). Some of these networks are Smart Home, Smart Office, Smart City, Smart Factory, among others. For this case study, the scenario is based on the connected objects in a Smart Home. The IoT Gateway's AI classifies the objects connected in the network into workgroups and roles by layer [34], and then, when an object requires resources, the IoT Gateway routes it, selecting the best node to provide them. The AI creates these groups to provide an automatic service to a user based on their features of functions, resources, and capacities. The following is a case of service within the Smart Home, e, g., when a visitor arrives at the house, the AI selects the objects with nearby features to provide this service. The system searches inside the house if there are users or not and automatically attends the door's visit. The necessary resources for this service are multimedia such as video and sound necessary to show the image of the visit on the objects that have this resource and that are close to the user inside the house. If there are no users inside the house, it will send them to objects or mobile devices in any location. One function that would be activated would be facial recognition and identity verification and then sent to be processed in the cloud. The first object to interact in the service would be the smart main door at the front of the house. This would be the requesting node for the resource and would execute the recognition and verification function. The IoT Gateway's AI would be in charge of distributing and transmitting the video and voice to the objective objects that have this resource and that meet the condition of being close to a user and with the capacity to reproduce it. Fig. 2 shows a case study for a workgroup that attends the door's visit service in a Smart Home through this architecture. This workgroup is organized according to the layers of the architecture in Fig. 1. When the smart door attends a visit, it activates a request to send video and image processing data. The image processing data (facial recognition) is sent through the AI interfaces until the visitor's identification is obtained. The video is sent to each connected object with multimedia playback functions that is close to the user and routed through the IoT Gateway. Therefore, the smart home announces a visit and the identification of a person located in the main door. If the user is away from home, the IoT Gateway will send the information through the cloud to the closest objects with multimedia functions to the user (E.g., Smart Car, Smartphone, Tablet).

Fig. 1 IoT Smart architecture



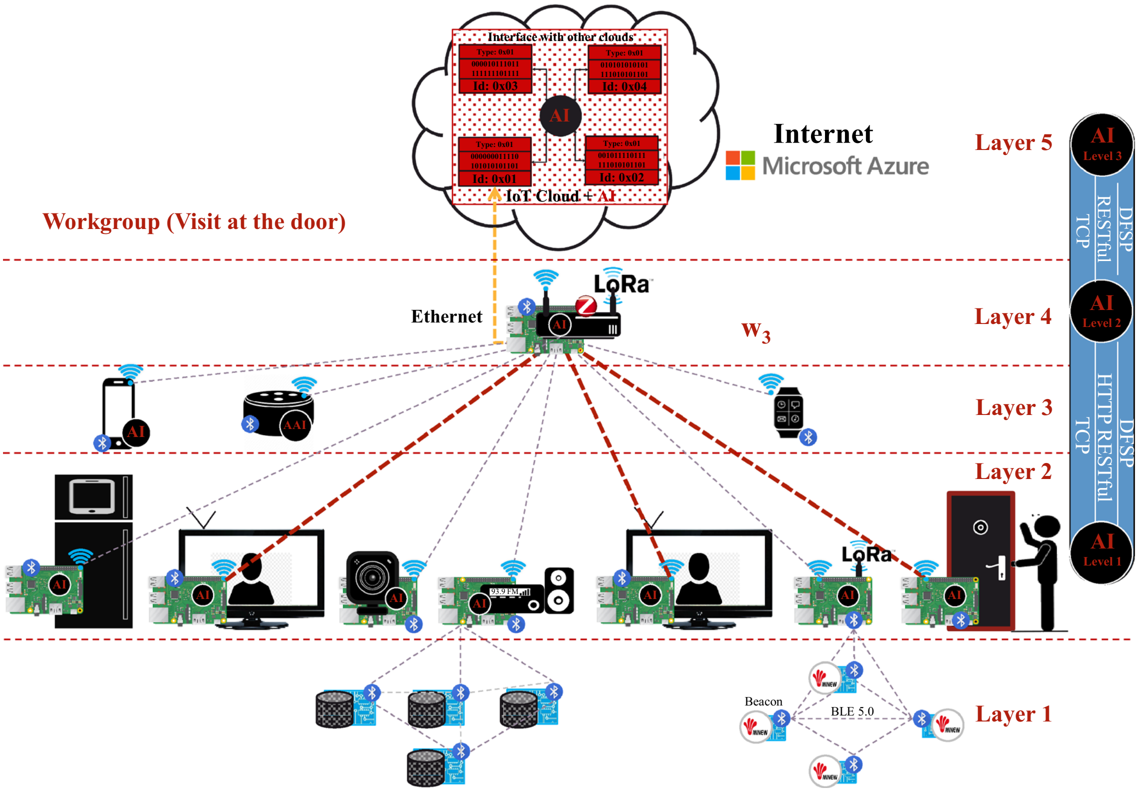


Fig. 2 Case study with a single workgroup

In Smart Home, with this architecture, the IoT Gateway, through the IoT nodes, can provide multimedia services. Through user requests given to the smart assistant placed in the IoT Gateway, either by voice commands or either by a smartphone application, the user can play their favorite music in the audio system on the distributed speakers in the house, watch movies-on-demand, or automatic record surveillance videos of the house.

3.2 Intelligent system architecture

Once the network architecture has been explained, the system architecture must be discussed. The aim of the system is to improve the QoE of the services in the Smart Home, focusing on multimedia services. In order to achieve that, the intelligent system must enable or ask the user to enable the services the user may want to use after other services and learn how to disrupt the user's activity the least number of times.

The intelligent system is composed of several modules, located in the IoT Gateway, that are interconnected to provide the desired functionality. The architecture of the system is depicted in Fig. 3.

The first module in the system is the recording module. This module is not an intelligent system, but it is the necessary first step to get the data the system needs. Each time the user consumes a service, the IoT Gateway records a data array with the structure shown in Table 1. Due to space constraints in the table, the user id column has been omitted and fields like the time, the options or the duration are shown as unique columns. However, in the dataset, they are split. For instance, the time and duration fields are split into day (only regarding the time column), hour, minute and second. As regards options, the entry is a set of columns, from option 1 to option 5, represented by binary values. The meaning of the options is meaningless for the system. Only the service finds this meaning useful. In Table 1, the first row means that the user enabled the heating service the second day of the week at 5:39 PM for 1 hour 42 min. Moreover, that day was a working day (the type of day field shows us this) and that service is not a multimedia service. The options here show us that the system was enabled in heat mode. The IoT Gateway manages equivalency tables to transform the meaning of these fields into their values. Consequently, the smart system works with integers and that makes easier the processing of the data. Table 2 shows an example of an equivalency table. The options used in this record can be non-exclusive options.

The second module of the system's architecture is the data preprocessing module. This is a software module that computes the data to transform it into the input of the next modules. The data preprocessing process is detailed in Sect. 4.

After the data has been processed, the datasets extracted from the logs provided by the record module are sent to the classifying module. Here we have to distinguish between the training phase and the prediction phase. In the training phase, the dataset extracted is used to train the classifying. Therefore, a training, validator and test dataset are extracted. Once the classifying system has been trained, the classifying module is used to predict the next service to be consumed. Consequently, the data sent by the data preprocessing module are the next inputs for the classification. In that case, the classifying module returns the predicted service.

The next module in the system is the RL module. The RL module receives information about the services from the data preprocessing module. This information is used to build the initial states and to calculate the required metrics. For instance, the RL module needs to know if a specific service is a multimedia service. When the classifying module predicts a service consumption, that prediction is an input for the RL module. The RL module chooses the best action, as explained in Sect. 5, and that action is the output of the RL module.

Finally, the IoT Gateway has an actuator module that performs the action chosen by the RL algorithm.

Fig. 3 Intelligent system architecture

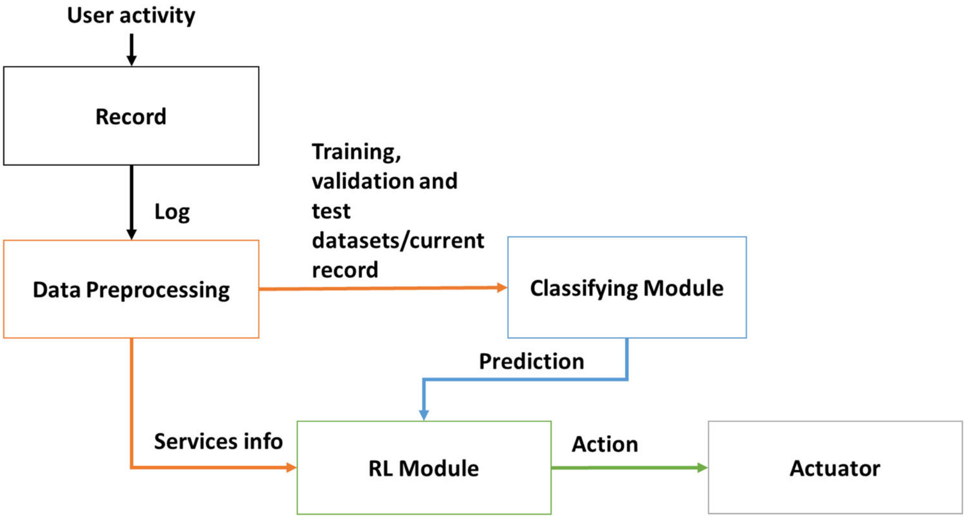


Table 1 Record data example

Time	Type of Day	Service	Description	Options	Duration	On/Off	Multimedia
2:17:39:17	1	2	Heat	10000	1:42:24	1	0
2:18:33:51	1	0	Lights	01001	0	0	0
3:9:11:25	0	3	Smart TV	10000	11:2:1	1	1

Table 2 Equivalency table example

Service	Option 1	Option 2	Option 3	Option 4	Option 5
0	Bedroom	Bathroom	Kitchen	Dining Room	Others
1	Main Door	Garage	Other Doors	Window 1	Window 2
2	Heat	Cool	Auto	Fan	Sleep Mode

In the next section, the data preprocessing and the classifying modules are described.

4 Preprocessing and classifying algorithms

In this section, the data preprocessing and the classifying modules are explained. First, the data preprocessing process is described. Its process and algorithms are detailed. Then, the classifying model chosen is described.

4.1 Data preprocessing process

Once the logs are provided to the preprocessing module, this module starts extracting some characteristics from them. Firstly, the RL module uses some data characteristics that are outputs of the preprocessing module. For the RL module, the statistical probability of changing from one state to another and the mean of the timestamp when it does are important data. Consequently, the preprocessing module transforms the data extracting these statistics. In order to achieve this, the module manages Markov chains. The definition of these chains adapted to the problem is described in (1):

$$\begin{aligned} P[X_{n+1} &= x_{n+1}|X_0 = x_0, X_1 = x_1, \ldots, X_n = X_n] \\ &= P[X_{n+1} = x_{n+1}|X_n = X_n] \end{aligned} \tag{1}$$

Where X_{n+1} is the next service consumed, X_n, the service consumed in the iteration number n and x_n is the service number n in the services set S.

In this system, the data will be processed regardless the time. That means, regarding the preprocessing, the time does not change the probability of the transition between services. That is depicted in (2):

$$P[X_{n+1} = j|X_n = i = P[X_1 = j|X_0 = i]\forall i,j \in S \tag{2}$$

The fact that the Markov chain does not consider time to set the probability does not mean that in the system the time is not considered an important input. However, the RL algorithm will use time in a different manner.

Once the records have been read, the preprocessing module turns them into matrices so that the RL algorithm can operate efficiently with them. The first data the RL will need is the probability of consuming a service. To set this probability, depending on the last service consumed, the preprocessing module builds a transition matrix. This matrix, given a consumed state i and a possible next consumed state j, defines the probability p of demanding j. The preprocessing module must satisfy the constraints defined in (3) and (4):

$$p(i,j) \geq 0 \quad \forall i,j \in S \tag{3}$$

$$\sum_{j \in S} p(i,j) = 1 \quad \forall i \in S \tag{4}$$

Where $p(i,j)$ is the probability of consuming the service j after consuming the service i.

In this system, the probability of consuming a service is not enough. Another important statistical data is the time between transitions. If the time is not considered, the system could ask for the right service hours before the user wants to enable it. This, although will not be treated as a feature of the definitions in the RL algorithm, will be necessary information to implement some of the actions of the RL system (see Sect. 5). For these calculations, the preprocessing module uses the time and duration of the records for each service. As regards the means of time and duration, and to take advantage of the incremental processing of records, the calculation will use a recursive formula. In order to calculate the mean, the equation defined in (5) is used.

$$\mu_n = \frac{(n-1) * \mu_{n-1+} t_n}{n} \tag{5}$$

Where μ_n is the media with n records, n is the number of records and t_n is the time or duration of the record number n. From this equation, we need to define the basic case as in (6):

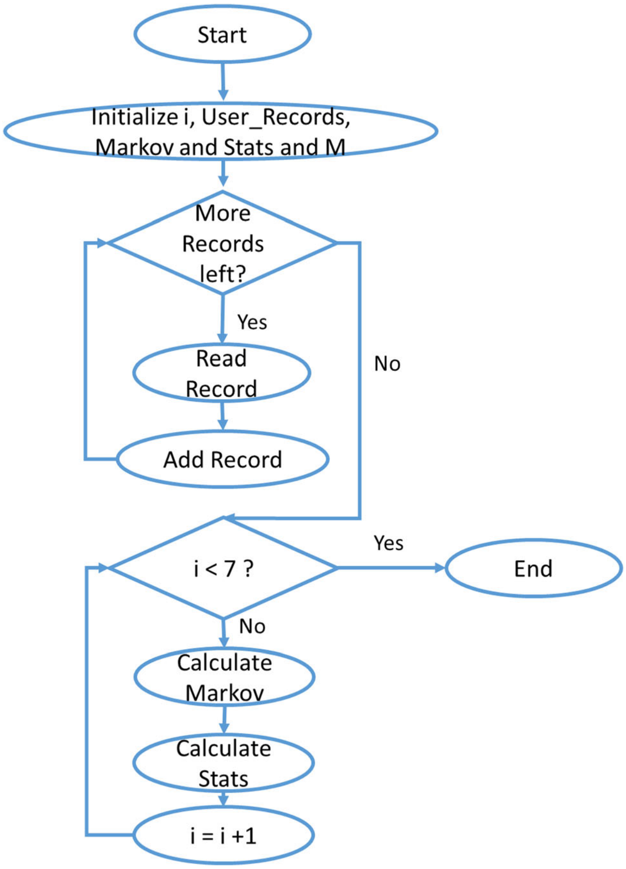


Fig. 4 Data preprocessing algorithm

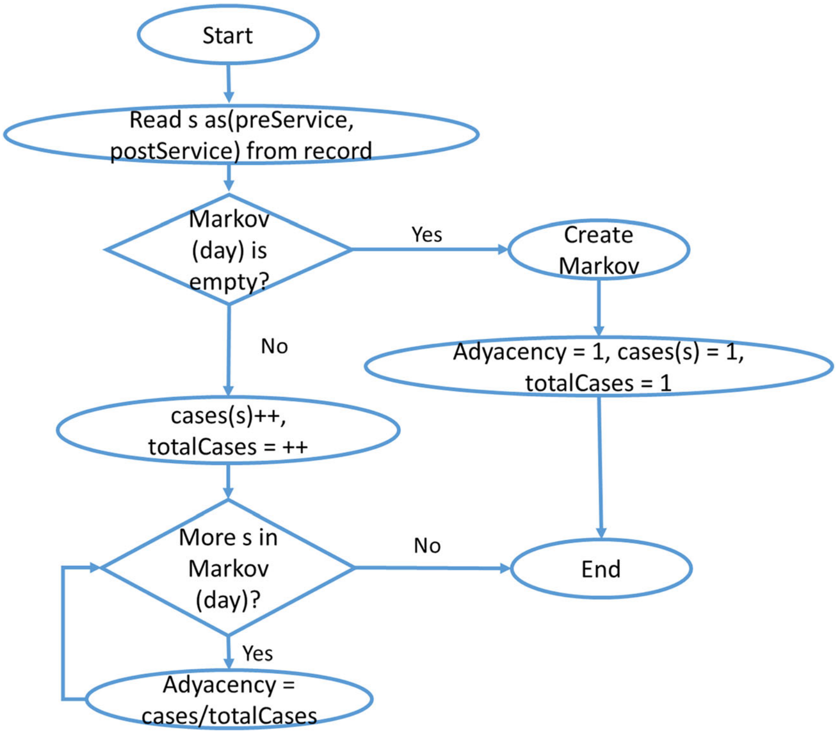
$$\mu_1 = t_0 \tag{6}$$

The variance is calculated in a similar way. The recursive formula described in (6) is used to avoid iterating through each past record when a new service consumption is ended.

$$s_n^2 = s_{n-1}^2 + \frac{(t_n - \mu_{n-1})^2}{n} - \frac{s_{n-1}^2}{n-1} \tag{7}$$

With these definitions, we can set the algorithm of the data preprocessing module. This algorithm is shown in Fig. 4, in a flow diagram that describes the algorithm. First, the data needed is initialized. Then, all the records are processed until there is no more records left. For each day of the week, that is why the next condition is compared with 7, the Markov and the stats are calculated. The pseudocode of the algorithm is described in Algorithm 1. This algorithm defines the main procedures of the data preprocessing module. Given a set of records from the IoT Gateway, the module processes the records to assign them to users and days (User_Records). Then, the Markov transition matrix (Markov) is calculated. This is done in an iterative way. The algorithm of the Markov data building is shown in Algorithm 2 and explained later. Then, the other statistics needed are extracted from the time and duration data. This subroutine is explained in Algorithm 3. Finally, the datasets for training and validating the classification module are extracted.

Fig. 5 Markov transition matrix calculation algorithm


Start
Read s as(preService, postService) from record
Markov (day) is empty?
Yes
Create Markov
Adyacency = 1, cases(s) = 1, totalCases = 1
No
cases(s)++, totalCases = ++
More s in Markov (day)?
No
End
Yes
Adyacency = cases/totalCases

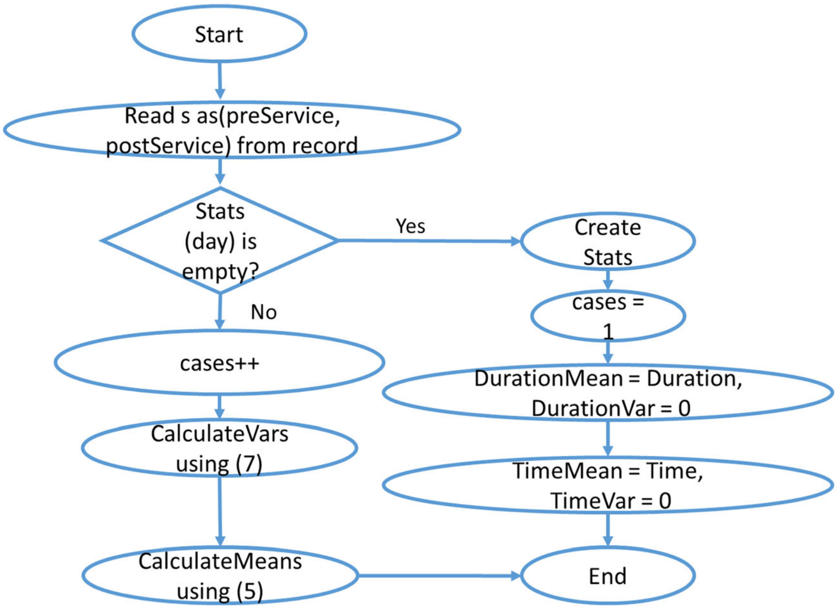

Start
Read s as(preService, postService) from record
Stats (day) is empty?
Yes
No
Create Stats
cases = 1
cases++
DurationMean = Duration, DurationVar = 0
CalculateVars using (7)
TimeMean = Time, TimeVar = 0
CalculateMeans using (5)
End

Fig. 6 Statistics calculation algorithm

As regards the Markov structure, it is calculated as it is shown in Fig 5, that depicts the flow diagram of this algorithm. In order to calculate the transition matrix, we need to know how many times a certain service is consumed after another one. We need to store the data in the Markov structure. In Fig. 5, it is shown how this info is read from the record. The structure is indexed based on the day, the first service and the second service, which is consumed after the first one. Moreover, for each first service, we need the total amount of transitions, totalCases. If we find a record and the structure is not created, we need to create it. And then, the totalCases is set to 1, as shown in Fig. 5. . If that service is the first time that is consumed, the following service consumed has a 100% of transitions. Otherwise, we need to iterate through all the previous transitions from that service to calculate the probability for each second service. That is shown in the last loop in Fig. 5. Algorithm 2 describes the same process with more detail.

Algorithm 1. Data Preprocessing

```
Given: Records //Ununstructed Records from the Gateway
Var User_Records, var Markov, var Stats

For each record in Records do
    If Get_User(record) not in User_Records
        Add_User_Record(User_Records, Get_User(record))
    End If
    Add_Record(User_Records, record)
End For

For each record in User_Recods
    For each day in range(1:7)
        daily_records = Get_Records_From_Day(day)
        For each d_record in daily_records
            Calculate_Markov(Markov, day, d_record)
            Calculate_Statistics(Stats, day, d_record)
        End For
    End For
End For
```

Table 3 Services description

Service 0	Service 1	Service 2	Service 3	Service 4	Service 5	Service 6
Lights	Doors	Heat	Smart Leisure	Music Player	Door Surveillance	Video Streaming

Finally, we find the statistics calculation in Fig. 6. Fig. 6 shows the flow diagram that corresponds to Algorithm 3. It depicts the two different ways of calculating the statistics. If there was no previous statistics for a specific day when the record is read, the data structure is created and initialized Otherwise, the statistics must be recalculated. By using equations (5) (6) and (7) these statistics can be recalculated each time a new service is consumed after another one in the same day or each time the new service ends its consumption (for calculating the duration).

Algorithm 2. Markov transition matrix calculation

```
Given: Markov, day, d_record
Var preService = d:record.service
Var postService = d.record.nextService

If Markov[day, preService, postService] is empty
    mark = Create_Markov(); //Create the structure
    mark.adyacency(preService, postService) = 1
    mark.service_cases(preService) = 1
    mark.totalCases = 1
Else
    mark = Markov[day, preService]
    mark.totalCases++
    mark[postService].cases++
    For each service in mark
        tempMark = mark[service]
        tempMark.adyacency = tempMark.cases/mark.totalCases
    End For
End If
```

In the following subsection, the classifying module is explained. That module uses some of the data provided by this module. In the training phase, the records are split to create the datasets. In the prediction phase, each new record is sent to the classifying module.

4.2 Classifying module

The classifying module is based on neural networks. Then, we have a deep learning method to predict the next service consumed by the user. The entries of the system will be the different measurements of the record (time, day, type of day, service consumed, duration and so on). In the records, we have 20 different features. As output, the different services provided. We define seven different services, shown in Table 3. A weak point of having a classifying system as the single method to predict service consumptions is that if a new service is added to the system, the classifying model has to be redefined and trained again. However, with an RL as a supervisor, that process can be delayed, and a provisional action can be added to avoid a QoE reduction until the classifying model is trained again.

The classifying model will be based on a neural network whose architecture is depicted in Fig. 7. We define a number of neurons in the input layer equals to the number of features in the data. In our case, we will have 20 features, so 20 different neurons in the input layer. The number of hidden layers, n_h, and the number of neurons in each hidden layer, m_h, will be parameters of the model. After testing the model varying these parameters, the model used will be the one with the highest accuracy. This evaluation will be presented in Sect. 6.

Algorithm 3. Statistics calculation

```
Given: Stats, day, d_record
Var preService = d:record.service
Var postService = d.record.nextService

If Stats[day, preService, postService] is empty
        stat = Create_Stats(); //Create the structure
        stat.meanDelta = postService.time-preService.time // (6)
        stat.varDelta = 0
        stat.meanDuration = preService.duration // (6)
        stat.varDuration = 0
        stat.totalCases = 1
Else
        stat = Stats[day, preService, postService]
        stat.totalCases++
        stat.varDelta += pow(postService.time - stat.meanDelta, 2)/stat.totalCases - stat.varDelta/(stat.totalCases-1) // (7)
        stat.meanDelta = (stat.totalCases-1)*stat.meanDelta + postService.time/ stat.totalCases // (5)
        stat.varDuration += pow(postService.duration - stat.meanDuration, 2)/stat.totalCases - stat.varDuration/(stat.totalCases-1) // (7)
        stat.meanDuration = (stat.totalCases-1)*stat.meanDuration + postService.duration/ stat.totalCases // (5)

End If
```

As regards the output layer, it will be composed of 7 neurons, one for each class to be detected by the system. If the number of different services in the Smart Home increases, the system would need to extract more relevant features and the model would need to be adapted. However, this adaptation is not going to be considered in this paper, and it will be considered as future work.

The loss function will be cross-entropy and the optimization method will be Root Mean Square Propagation

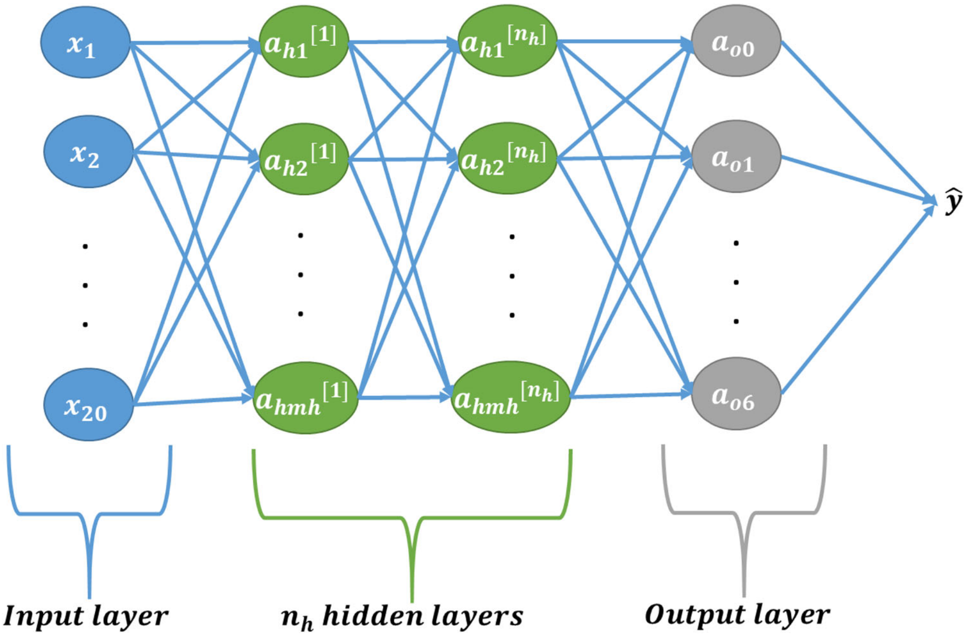


Fig. 7 Classifying neural network architecture

(RMSprop), which achieves good results in multi-layer neural networks [36].

Another aspect of the model that has to be chosen is the activation function. We will use the ReLU function as the activation function of the hidden layer, a_h [37]. The softmax function will be used as the activation function of the output layer, a_o, to get the probabilities of belonging to each class [38].

With the model presented, the next service consumed by the user can be predicted. In the next subsection, we define the RL module, that will choose the best action to perform based on this prediction.

5 Reinforcement learning module

This section describes the RL algorithm that will be used to reduce the impact of bad predictions. First, the environment, the states and the actions are defined. Then, the data structures that the algorithm needs to work are described. Finally, the rewards calculation and the policy of the system are detailed.

5.1 Environment, states and actions

In this subsection, the environment, the states and the actions that the algorithm will use to implement the reinforcement learning will be described.

Firstly, we have to define the environment. The reinforcement learning will act in the Smart Home environment. Exactly, it will notify the IoT Gateway which command must perform. Therefore, the agent of the reinforcement learning algorithm will be the IoT Gateway. Initially, it could be thought that the user would be the agent, but the user will be only the source of information of the actions performed by the agent. For instance, when the user consumes a service and the algorithm decides to start another service after a certain amount of time, the user may not need that service and give the order to turn it off. The user, then, is an input for the algorithm. In this case, the user is saying that the action performed was not chosen correctly. This fact would affect the reward of the action. We will discuss that in the next subsections. For now, it is important to notice that the user will be an input, not the agent that modifies the environment. Despite this fact, the user is the one who starts enabling the services. Consequently, if the definition of the states would be only based on the current service running in the Smart Home, the definition of the environment and the agent would be more complicated. Furthermore, the algorithm would be incomplete, due to the fact that we need to know which service is supposed to be necessary to activate. It is there when we need the classifying module. Moreover, due to the user patterns, the day and the current time are also relevant data.

From the previous discussion, the following definition of the states is obtained. The RL algorithm will decide which action needs to be performed based on the current service consumed, the next service predicted and the current day. This presents a problem that will not be addressed in this paper: how new states are generated through the use of the system. We will consider in this paper that the states are statics and are generated from the transition matrix obtained from the preprocessing module. However, new patterns can be adopted by the user and this will be discussed in future works.

Usually, the states of an RL algorithm can be depicted in a state diagram. However, in this scenario, the state diagram can be composed of a high number of states, depending on the number of services. This can make the algorithm not scalable. Nevertheless, the Smart Home environments do not have a high number of services. Furthermore, not all the services can be important enough to define a state. In this paper, we will use all of them but, in future works, a categorization of services can be proposed to reduce the space of states.

Fig. 8 shows an example of a diagram of states with three different services to enable. In this case, there is no probability of transitioning from state 2 (S2) to state 3 (S3). Therefore, there is no transition between those states in the graph. Furthermore, depending on whether the user actually consumes the predicted service, the next state might be with the same service being consumed. That may happen when the classifying module does not predict the next service accurately or when the prediction was to consume again the same service. The state diagram is not, then, a direct representation of the Markov chain derived from the transition matrix. There are transitions to states with the same current service that do not represent the same service being consumed twice. In Fig 8, in order to make easier the readability of the diagram, the states are composed only of

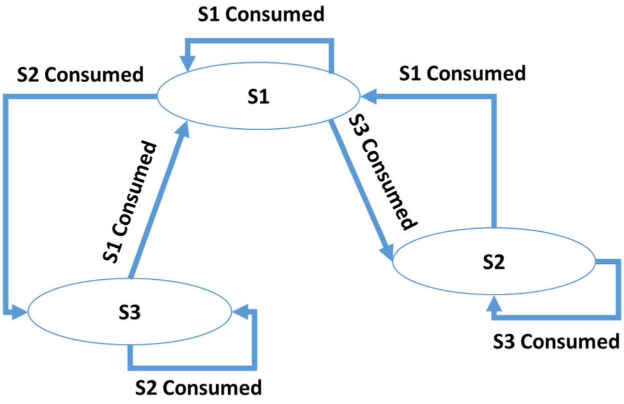


Fig. 8 State diagram with three services

the current service. However, the diagram is more complex because the algorithm defines the state as a pair of states. The first state is the current state and the second one is the predicted state. Fig. 9 depicts the diagram of states of the state S2 of Fig. 8. Within each state of Fig. 8, there would be a subdiagram with similar transitions. Although this can add some difficulty to understand the algorithm, the number of states is finite and not big enough to present a deep-learning algorithm for the policy function as in [39]. However, it could be a future work to study.

For each state defined, a set of actions can be performed. The actions, however, must be classified based on how much they interrupt the user's activity. This helps the algorithm to not reduce the user's QoE as much if the predictor fails to predict the next service. For this proposal, six different actions are going to be defined and classified depending on this intrusion level. The actions are shown in Table 4, where the level of intrusion and the description are detailed. The simplest action is to wait, without executing any service. This has the lowest impact on the user because they do not have to do anything. However, this does not mean it is always the best action, because sometimes enabling a service or turning on a node can reduce the waiting time for the user or implying other advantages (for instance, saving energy or heating up the house before the user arrives). An alternative with low impact would be to ask if the user wants to enable the predicted service or another one from a list of similar alternatives (services from the same group). This is not too different from a manual service selection. If the system only asks to enable a certain service or it turns on a node it will have a higher intrusion from the user's point of view. We have to take into account that certain services are more intrusive than others. This is reflected in the actions. The system can automatically start a predicted service, with some options or subfunctionalities that are not too intrusive, such as perform a search on Internet or increase or decrease the temperature a few degrees. However, other services, like opening a door, turning on the TV or start playing a song or video have a bigger impact on the user. Finally, stopping services that the user is using or turning off nodes automatically has the biggest impact on the user, so transitions that require disable services will have the biggest level.

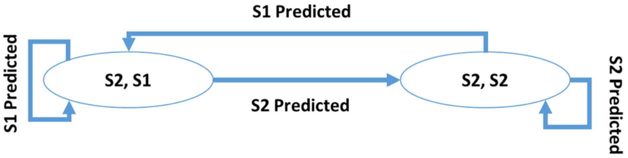


Fig. 9 Subdiagram of state S2

Table 4 Actions

Level	Action
0	Wait
1	Ask with alternatives
2	Enable a node or ask to start a service
3	Start a low-impact service
4	Start a high-impact service
5	Stop services or turn off a node

The definitions of the actions are not simple actions as they could be found in other RL algorithms, but there are actions that will depend on the predicted service. That means, that enabling a node will enable a node that provides the predicted service, starting a service will start the service predicted or a module of the service and so on.

These actions provoke changes in the network performance, and, depending on the next values obtained from the Statistic Analyzer, the reward value of the action taken will be updated.

The way the rewards are assigned and calculated and how the actions are performed are described in the next subsections. However, we first need to know the data structures and concepts.

5.2 Data structure

In this subsection, the data structures that the RL algorithm uses are described.

First of all, the RL algorithm needs a structure where the reward of each action for each state is stored. This data structure will be a matrix, where the rows will be the states and the columns will be the different actions of the algorithm. In this case, the states are a pair of current service and predicted service values. Table 5 shows an example of the data structure for three different services, following the same state diagram that is shown in Fig. 8.

Secondly, the algorithm will need information about the services. This information is given by the data

Table 5 Reward-state matrix

State	Action 1	Action 2	Action 3
S1, S1	1.3	7.5	2.21
S1, S2	0.4	1.2	2.05
S1, S3	1	2	0.75
S2, S1	1	2	0.75
S2, S2	1	2	0.75
S3, S1	1	2	0.75
S3, S3	1	2	0.75

Table 6 InfoServices matrix example

Service	Workgroup	Multimedia	Impact
1	1	0	5
3	2	1	4
4	3	1	4

Table 7 Input user matrix example

Action	User option 1	User option 2	User option 3
a_1	0.2	0.25	0.1
a_2	0.4	0.8	0.1
a_3	1	0	0

preprocessing module. For each service, the following data is required: the group where the service belongs, the impact that the service has on the user and if the service is multimedia. The group of the service is data that the IoT Gateway knows because it is that agent who categorizes the services attending to the architecture presented in Sect. 3. If the service is multimedia it also comes from the IoT Gateway. Like the group of the service (workgroup), this is information that is presented in the dataset. However, the last data that is needed, the impact of the service on the user, must be set by the RL algorithm. A simpler classifying module could be added in the system architecture just before the RL module to determine which services have a bigger impact on the user. However, to make the proposal simpler, the IoT Gateway provides the RL module with this information, obtained statically from the group of the service. Table 6 shows an example of this data structure, which we will name as infoServices. The impact field is a positive integer. The bigger the impact value, the more intrusive the service is.

The last data structure needed to implement this RL algorithm is the Input User Matrix. This data structure will store how much a user input is needed for each action. That means, that the algorithm will be able to know the impact of having a certain input for the user for each action chosen. This will be useful for knowing how much a certain action was appropriate for a state, that is, how the RL algorithm should be rewarded. For instance, if a set of services is given to the user to choose which one should be started, if the user selects the predicted, the reward will be different from the one obtained if the user discards all the possibilities. This will be discussed in more detail in the next subsection, where the policy of the RL algorithm is detailed. Table 7 shows an example of this matrix.

5.3 Rewards, policy and objective function

In order to define the reward properly, we have to analyze the problem we are dealing with, because it is the problem, and the scenario, the one that defines the appropriate method to obtain and calculate the reward. The reward will indicate to the system whether the action chosen was effective in the same state or, otherwise, if there are better options. If reinforcement learning is applied to a game, the effects that a performed action over the player determines the reward. If the game provides an actual reward such as finishing a level or defeating an enemy, the reward will be positive. If other actions usually drive to losing a life, losing objects and so on, the reward will be lower and will decrease if we choose that action, even being able to contain a negative value.

In the environment previously defined, the goal of the system is to avoid the user from enabling services or nodes. However, it is also important to reduce the intrusion of the system, especially regarding multimedia services. Therefore, the input of the user will be quite important to know if the action chosen by the system was appropriate.

Applying the RL system to the result of a classifying algorithm modifies the way the reward is calculated. In this case, the reward will be a measure of the number of times the classifying module has predicted correctly which service would be consumed. Therefore, if the module provides high accuracy with certain services, the actions with a high level of intrusion can work well. On the other hand, when it has low accuracy, performing an action with a low level of intrusion could be a better option.

The algorithm has a set of actions, J. We are going to describe the general case where several actions can belong to the same level classification. However, this definition will also be valid when there can be only one action per level. Consequently, we can define the reward obtained after choosing an action j as in (8).

$$r_j = \ u, \ where \tag{8}$$

$$u = 1 \leftrightarrow Input(User) < \beta \tag{9}$$

$$u = -1 \leftrightarrow Input(User) \geq \beta \tag{10}$$

where Input(User) is a parameter that returns the IoT Gateway and its value depends on the action chosen and the action the user does after. The possible values were defined previously, as an example, in Table 7. Moreover, β is a threshold defined to classify the action performed by the user as corrective or not. Then, if the action was corrective, the reward should be decreased.

Nonetheless, if we only consider the reward like that, we can drive the system to a point where the actions of less level, due to the fact that they are less intrusive, present a higher reward. In order to solve this, the reward should be incremented regarding the level of the action chosen. Consequently, the reward given to an action j in the iteration i will be defined by (11).

$$r_{i,j} = u_{i,j} * \frac{level(j)}{Max(level)} \tag{11}$$

Where level(j) is the level of the action j chosen.

The last adjustment that needs to be done is to give extra importance to multimedia services. That means, for those actions coming from a state whose current service is multimedia, the reward should increase or decrease at a higher rate. In (12) we can see this adjustment.

$$r_{i,j} = (1 + multimedia(s)) * u_i * \frac{level}{Max(level)} \tag{12}$$

Where $multimedia(s)$ is the flag in the dataset that identifies the service s as a multimedia service.

We can define, then, the total reward of an action j as (13).

$$R_j = \sum_{i=0}^{n} r_i \quad where \quad R_j \geq 0 \forall\, j \in J \tag{13}$$

Initially, the rewards are calculated depending on the level and the probability of transitioning from the initial state S_a to S_b as is defined in (14):

$$\begin{aligned} r_{0,j} = {} & (1 + multimedia) * (P(S_a, S_b) * \frac{level(j)}{Max(level)} \\ & + (1 - P(S_a, S_b)) * \frac{Max(level) - level(j)}{Max(level)}) \end{aligned} \tag{14}$$

We need to define then the policy of the system. In subsection 5.2 we defined a matrix that contains the rewards for each pair of state and action. Usually, RL algorithms choose the action with the highest reward. However, in this case, the policy will have a component of exploration. This component will force the algorithm to try actions that had not been tried for a high number of iterations. Equation (15) defines the function ρ that represents the probability of choosing and action $\boldsymbol{\alpha}$ that has less reward than the action with the maximum reward, n, for the same state.

$$\rho_\alpha = \frac{R_n - R_\alpha}{|level(n) - level(\alpha)|} + \omega * i \tag{15}$$

Where i is the number of iterations that have passed since the last time action α was chosen and ω is a parameter that sets a weight to the exploration.

The policy function can be described then by (16):

$$\pi(s) = a_j | \rho_j \geq \rho_n \quad \forall\, j,\ n \in J \tag{16}$$

6 Methodology and results

In this section, the tests performed and the results obtained are discussed. First, the classifying model will be tested in order to obtain goods parameters. Then, the QoE will be measured to check if the introduction of an RL algorithm improves it.

6.1 Classification analysis

In Sect. 4, the classifying model was defined based on two parameters. These parameters, the number of hidden layers n_h, and the number of neurons in each hidden layer m_h, are analyzed in this subsection. In order to get the better values for these parameters, the model will be trained, given the same dataset, for the values of n_h, and m_h shown in Table 8.

The results of the evaluation of the model for each pair of parameters values are shown in Fig. 10. In the figure, the accuracy for each number of neurons is shown for each model with a different number of hidden layers. The size of the training dataset corresponds to two weeks of user activity. Although the mean of the values of each series could be interesting for other experiments, we only need

Table 8 Parameters for Classification analysis

Parameter	Val 1	Val 2	Val 3	Val 4	Val 5	Val 6	Val 7
n_h	1	2	3	4	5	–	–
m_h	1	2	5	10	12	15	20

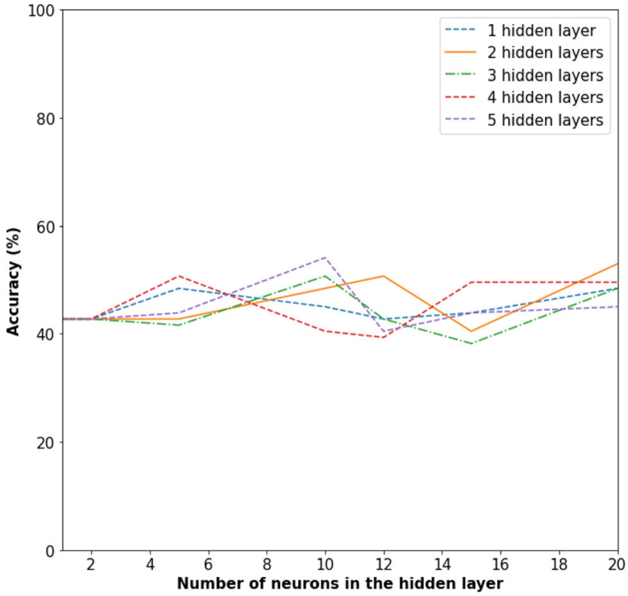


Fig. 10 Classification model results

here the maximum value of all the series to select the values of the parameters with the best accuracy. In this case, the model works better with this kind of data when it has five hidden layers and ten neurons per hidden layer. With those values, the accuracy of the classifying model is 54%. Despite the fact the accuracy might be improved with further feature selection or with larger datasets, this is a good accuracy to prove the hypothesis of this work.

Furthermore, the results of the deep learning model are tested against another classifying algorithm. In this case, we chose the KNN algorithm to implement a classifying algorithm that predicts the next service. In this case, the KNN algorithm is parametrized depending on the number of neighbors. Fig. 11 shows the results obtained from the KNN algorithm for a number of neighbors from 1 to 18. The maximum obtained accuracy, with the same dataset that the deep learning model, is 41.21% with three neighbors.

Fig. 12 shows an example of how preparing the data from the Smart Home environment may help to get good levels of accuracy. The figure depicts the violin graph of the feature "type of day". This feature has two values: 1 if the day is a working day 0 it is not. Since the services with higher ID are the services regarding multimedia and leisure (see Table 3), the probability of having a high value in service is higher when the type of day is 0.

Finally, Table 9 shows the running time of the different parts of the algorithm. We have measured the mean of the running times from the different parts of the proposal. The experiments were carried on an AMD Ryzen 5 5600X with 32GM RAM DDR4. The results show that the times fit the environment restrictions, which are not extremely demanding in terms of running time.

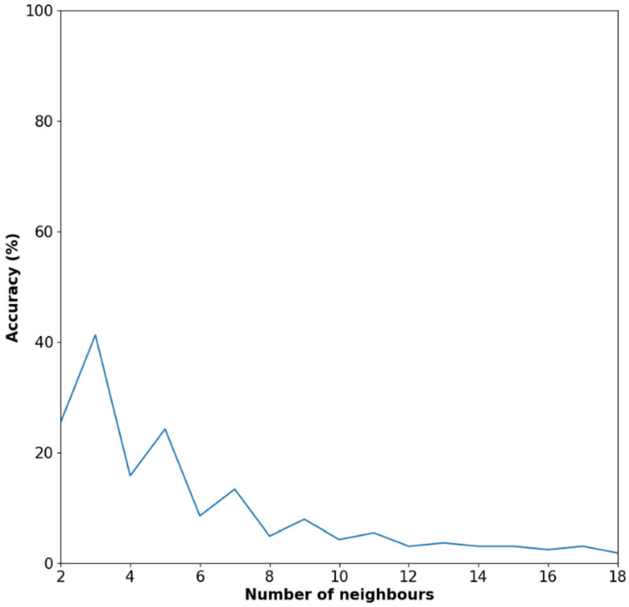


Fig. 11 KNN classification results

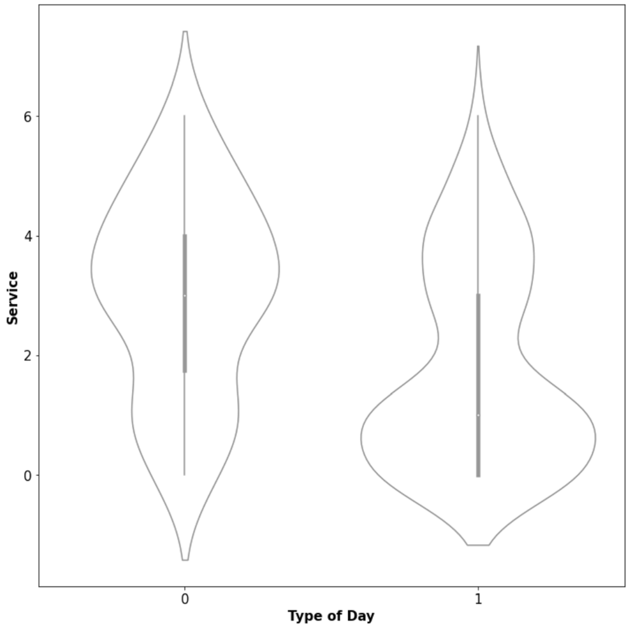


Fig. 12 Violin graph of service depending on the type of day

6.2 QoE in smart home results

In this section, the most important metric for this work, QoE at Smart Home, will be calculated for a KNN classifying system, the deep learning system and the deep learning system with the RL as a supervisor.

However, we need to define first how we calculate QoE applied for the Smart Home scenario. At Smart Home, unlike in multimedia scenarios, the delay, bandwidth and jitter will not be our concern. We are going to put aside the network resources and focus on the user. Thereby, the QoE decreases when the user is interrupted with consults about services they do not want to use. Moreover, if the system starts a service the user does not want to consume or to stop a service the user wants to keep consuming, the QoE will also be decreased. After each prediction, the QoE will be updated based on the loss using the function λ (17).

$$\lambda = L(y, \hat{y}) * (1 + multimedia) \tag{17}$$

Where $L(y,\hat{y}) = \begin{cases} 0 & if\ y = \hat{y} \\ 1 & if\ y \neq\ \hat{y} \end{cases}$, and multimedia refers to the current service.

Note that the loss defined is only applied to the classifying. For the RL algorithm, the loss is defined by the user input matrix, depending on the action chosen.

Table 9 Running time

	Data Preprocessing	Classifficator (KNN)	Classifcator(NN)	Reinforcement learning prediction
Running time(ms)	24	9	21	1

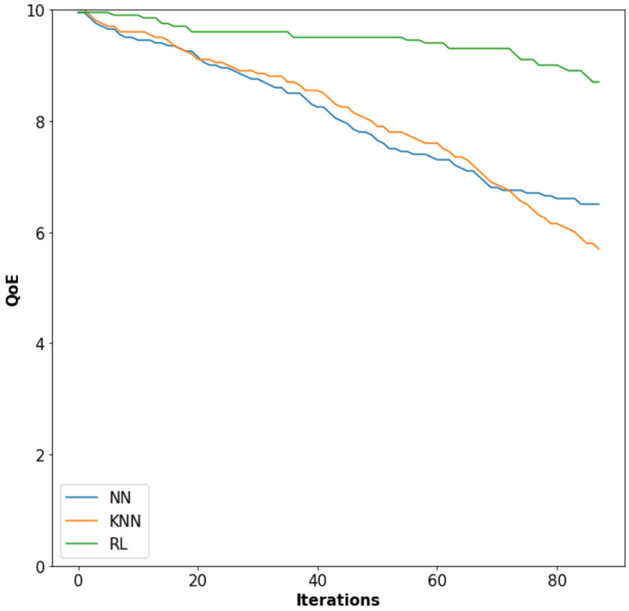


Fig. 13 QoE after 88 iterations with $\alpha = 20$

Therefore, let α be a parameter to weight the decreases, the total QoE after the n iterations (transitions) of the system is determined by (18).

$$QoE(n) = Max(QoE) - \frac{\sum_{i=0}^{n} \lambda_i}{\alpha} \tag{18}$$

In order to maintain similarity with the QoE defined in multimedia, the max value of QoE in our experiments will be 10. As regards α, it can be adjusted depending on the length of the measurement. We will show three different experiments where the parameter α is different, but the number of iterations does not vary. The number of iterations will be 88 for each measurement.

Fig. 13 depicts the QoE obtained during all the iterations with an α value of 20. With this α value, the RL algorithm obtains a QoE of 8.7. the deep learning algorithm obtained a QoE of 6.56 and the KNN algorithm of 5.89. Although the RL obtains a high QoE, the other two methods are not too far from each other in terms of QoE. Despite the good QoE obtained with the classifying methods without RL, the accuracy of the predictions has been low. This α value only shows a real behavior when the user cares less about interruptions.

The QoE values obtained with an α value of 15 are shown in Fig. 14. Again, the RL algorithm obtains better QoE than the other alternatives, with an 8.26. In this experiment, the deep learning algorithm obtains a 5.53 of QoE. The KNN algorithm is again the one that provides the worst performance with a QoE of 4.26. These results might represent with more accuracy the average user, whose QoE decreases with the interruptions. In addition, the RL presents the least decrease with the change of the α value.

Finally, Fig. 15 illustrates the results when the parameter α is reduced to 10. In this case, the bad predictions that have as a consequence to interrupt ongoing services decrease quickly the QoE. The RL supervisor, by choosing actions that prevent interruptions when the classifying algorithm does not give an accurate prediction, presents the best QoE with 7.39. One more time, the deep learning classifying model presents better results than KNN, with 3.59 and 1.53, respectively. The decreased pace of the QoE, in this case, may show an unrealistic situation where the user punishes the system too much. However, it shows how the RL algorithm improves the performance of the system.

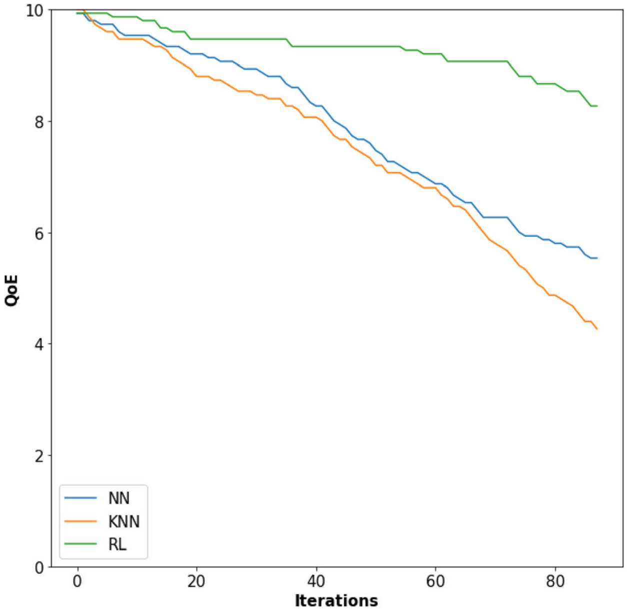


Fig. 14 QoE after 88 iterations with $\alpha = 15$

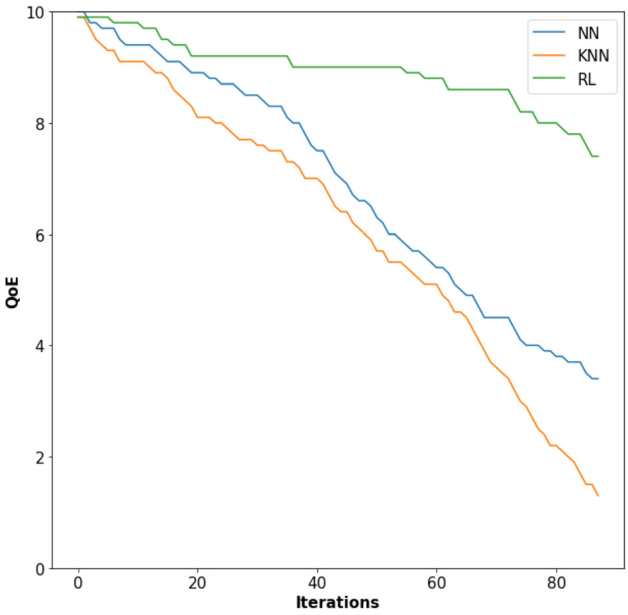


Fig. 15 QoE after 88 iterations with $\alpha = 10$

7 Conclusion and future work

IoT has provided new ways of networking, communicating and sensor development. In addition, with its introduction, several new applications have been designed in several fields. Smart Home is a new way of understanding services at home. With the communication that this technology offers, users can access new services at home. In addition, AI can help automatizing tasks at home using these services and architectures.

In this paper, we have introduced an intelligent system to automatize Smart Home services management. The aim of the intelligent system was to avoid user interruptions to guarantee a good QoE, prioritizing the multimedia services. In the system, we have designed the data preprocessing process, the classifying algorithm and the RL system that improves the performance, defining all the related concepts needed to describe the scenario and so that the system can interact and provide functionality.

We have also defined QoE for this scenario and we have measured it for different parameters and systems.

Results show that the deep learning classifying model proposed achieves better accuracy than other algorithms like KNN, improving their performance around 33%. The QoE of the deep learning algorithm shown with different values of the parameters has always been higher than the KNN algorithm. With high values of the parameter α, the QoE obtained is higher, being in the experiment 5.53 for the deep learning algorithm and 4.26 for the KNN. This difference, based on the difference in accuracy, gets higher when the weights of the prediction increase. That is when α has lower values. This difference raised up until 2.06, with QoE values of 3.59 for deep learning and 1.53 for KNN. Nevertheless, the most remarkable results are the ones obtained from the RL system, which manages the predictions to reduce the impact and obtains QoE values of 8.7 when α has the highest value and 7.39 when it has the lowest. That makes a difference of 3.17 in the first case and 3.8 in the second one. That shows that the inclusion of an RL system improves the QoE in Smart Home environments when the classifying cannot predict the next service with high accuracy.

There are several aspects that can be studied in future works. Firstly, the system can be evaluated against better classifying models. That would include the study of improving the accuracy obtained in this work. The feature selection could be enhanced, reducing dimensionality [40] or an automatized label system could be included in the system [41]. Other parameters, such as the activation function or the optimization parameters could be changed to improve the accuracy. The scalability of the system could be studied. Mechanisms to adapt the system to new services could be defined to efficiently control a bigger number of states. Moreover, the system could be compared to other classifying models, besides KNN. The QoE metric could add an intelligent method to select the α parameter based on the user profile. Thereby, some actions to correct low QoE could be included in the system. Finally, this system could be adapted to face other problems in IoT, such as video surveillance or Industrial IoT environments, working together with other solutions [42] or being integrated in other netwokrs like 5G [43].

Authors contribution The authors declare that each author has contributed to the realization of this work.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work has been partially supported by the "Ministerio de Economía y Competitividad" in the "Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento" within the project under Grant TIN2017-84802-C2-1-P. This work has also been partially founded by the Universitat Politècnica de València through the post-doctoral PAID-10-20 program.

Data availability The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Declarations

Conflict of interest The authors declare there were no conflicts of interest.

References

1. Smart utilities should discover smart homes. https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2017/may/Smart_Utilities_Should_Discover_Smart_Homes.pdf. Accessed 1 Jan 2021.
2. Osservatorio domotica e smart things. https://www.trovaprezzi.it/magazine/trend/osservatorio-domotica-e-smart-things. Accessed 1 Feb 2021.
3. Garcia, M., Canovas, A., Edo, M., Lloret, J.: A. QoE management system for ubiquitous IPTV devices, The Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM 2009. Sliema, Malta (2009)
4. Vasicek, D., Jalowiczor, J., Sevcik, L., Voznak, M.: 2018 26th Telecommunications Forum (TELFOR). pp. 20-21. Belgrade, Serbia (2018). https://doi.org/10.1109/TELFOR.2018.8612078
5. Jie, Y., Pei, J.Y., Jun, L., Yun, G., Wei, X.: Smart Home System Based on IOT Technologies. 2013 International Conference on Computational and Information Sciences. pp. 21-23. Shiyang, China (2013). https://doi.org/10.1109/ICCIS.2013.468
6. Khan, A., Al-Zahrani, A., Al-Harbi, S., Al-Nashri, S., Khan, I.A.: 2018 15th Learning and Technology Conference (L&T). pp. 25-26. Jeddah, Saudi Arabia. (2018). https://doi.org/10.1109/LT.2018.8368484
7. Malche, T., Maheshwary, P.: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). pp. 10-11. Palladam, India (2017). https://doi.org/10.1109/I-SMAC.2017.8058258
8. Yang, H., Lee, W., Lee, H.: IoT Smart Home Adoption: The Importance of Proper Level Automation. J. Sensors. **2018**, 11, Article ID 6464036. https://doi.org/10.1155/2018/6464036
9. Risteska, B.L., Stojkoska, K.V., Trivodaliev. : A review of Internet of Things for smart home: Challenges and solutions. J. Clean. Prod. **140**, 1454–1464 (2017). https://doi.org/10.1016/j.jclepro.2016.10.006
10. Mussab Alaa, A.A., Zaidan, B.B.Z., MohammedTalal, M.L.M.K.: A review of smart home applications based on Internet of Things. J. Netw. Comput. Appl. **97**, 48–65 (2017). https://doi.org/10.1016/j.jnca.2017.08.017
11. Kuzlu, M., Pipattanasomporn, M., Rahman, S. Review of communication technologies for smart homes, building applications. 2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA). pp. 3–6. Bangkok, Thailand (2015). https://doi.org/10.1109/ISGT-Asia.2015.7437036
12. Kamel, E., Memari, A.M.: State-of-the-art review of energy smart homes. J. Archit. Eng. (2019). https://doi.org/10.1061/%28ASCE%29AE.1943-5568.0000337
13. Philip, N.Y., Rodrigues, J.J.P.C., Wang, H., Fong, S., Chen, J.: Internet of things for in-home health monitoring systems: current advances, challenges and future directions. IEEE J. Select. Areas Commun. (JSAC) **39**(2), 300–310 (2021). https://doi.org/10.1109/JSAC.2020.3042421
14. Apthorpe, N., Reisman, D., Feamster, N.: A smart home is no castle: privacy vulnerabilities of encrypted iot traffic. (2017). https://arXiv.org/1705.06805
15. Augusto-Gonzalez, J., Collen, A., Evangelatos, S., Anagnostopoulos, M., Spathoulas, G., Giannoutakis, K.M., Votis, K., Tzovaras, D., Genge, B., Gelenbe, E., Nijdam, N. A.: From internet of threats to internet of things: a cyber security architecture for smart homes. 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). Limassol, Cyprus (2019). https://doi.org/10.1109/CAMAD.2019.8858493
16. Lin, H., Bergmann, N.W.: IoT privacy and security challenges for smart home environments. Information **7**(3), 44 (2016). https://doi.org/10.3390/info7030044
17. Meng, Y., Zhang, W., Zhu, H., Shen, X.S.: Securing consumer IoT in the smart home: architecture, challenges, and countermeasures. IEEE Wireless Communications. **25**(6), 53–59 (2019). https://doi.org/10.1109/MWC.2017.1800100
18. Ammi, M., Alarabi, S., Benkhelifa, E.: Customized blockchain-based architecture for secure smart home for lightweight IoT. Inform. Process. Manag. (2021). https://doi.org/10.1016/j.ipm.2020.102482]
19. Atat, R., Liu, L., Jinsong, Wu., Li, G., Ye, C., Yang, Yi.: Big data meet cyber-physical systems: a panoramic survey. IEEE Access. **6**, 73603–73636 (2018). https://doi.org/10.1109/ACCESS.2018.2878681
20. Collotta, M., Pau, G.: A novel energy management approach for smart homes using bluetooth low energy. IEEE J. Sel. Areas Commun. **33**(12), 2988–2996 (2015). https://doi.org/10.1109/JSAC.2015.2481203
21. Learn about bluetooth. bluetooth Radio Versions. https://www.bluetooth.com/learn-about-bluetooth/radio-versions/. Accessed 25 Jan 2020
22. Al-Ali, A.R., Zualkernan, I.A., Rashid, M., Gupta, R., AliKarar, M.: A smart home energy management system using IoT and big data analytics approach. IEEE Transactions on Consumer Electronics. **63**(4), 426-434 (2017). https://doi.org/10.1109/TCE.2017.015014
23. Xia, C., Li, W., Chang, X., Delicato, F. C., Yang, T., Zomaya, A.L.: Edge-based Energy Management for Smart Homes. 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). Athens, Greece. (2018). https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00-19
24. Celik, B., Roche, R., Suryanarayanan, S., Bouquain, D., Miraoui, A.: Electric energy management in residential areas through coordination of multiple smart homes. Renew. Sustain. Energy Rev. **80**, 260–275 (2017). https://doi.org/10.1016/j.rser.2017.05.118
25. Jinsong, Wu., Guo, S., Li, J., Zeng, D.: Big data meet green challenges: big data toward green applications. IEEE Syst. J. **10**(3), 888–900 (2016). https://doi.org/10.1109/JSYST.2016.2550530
26. Amjad A., Rabby, F., Sadia, S., Patwary, M., Benkhelifa, E.: Cognitive edge computing based resource allocation framework for Internet of Things, 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC). Valencia, Spain (2017). https://doi.org/10.1109/FMEC.2017.7946430
27. Jararweh, Y., Al-Ayyoub, M.: Du'a Al-Zoubi, Elhadj Benkhelifa, An experimental framework for future smart cities using data fusion and software defined systems: the case of environmental monitoring for smart healthcare. Futur. Gener. Comput. Syst. **107**, 883–897 (2020). https://doi.org/10.1016/j.future.2018.01.038
28. Sodhro, A.H., Gurtov, A, Zahid, N, Pirbhulal, S, Wang, W, Ur Rahman, M.M., Imran, M.A., Abbasi, Q.H.: Towards convergence of AI and IoT for energy efficient communication in smart

homes. IEEE Internet Things J. (Early Access). pp. 1–1. https://doi.org/10.1109/JIOT.2020.3023667

29. Guo, X., Shen, Z., Zhang, Y., Teng, Wu.: Review on the application of artificial intelligence in smart homes. Smart Cities. **2**(3), 402–420 (2019). https://doi.org/10.3390/smartcities2030025
30. Sepasgozar, S., Karimi, R., Farahzadi, L., Moezzi, F., Shirowzhan, S., Ebrahimzadeh, S.M., Hui, F., Aye, L.: A systematic content review of artificial intelligence and the internet of things applications in smart home. Appl. Sci. **10**(9), 3074 (2020). https://doi.org/10.3390/app10093074
31. Song, H., Bai, J., Yi, Y., Jinsong, Wu., Liu, L.: Artificial intelligence enabled internet of things: network architecture and spectrum access. IEEE Comput. Intell. Mag. **15**(1), 44–51 (2020). https://doi.org/10.1109/MCI.2019.2954643
32. Lloret, J., Canovas, A., Sendra, S., Parra, L.: A smart communication architecture for ambient assisted living. IEEE Commun. Mag. **53**(1), 26–33 (2015)
33. Sahoo, S.R., Gupta, B.B.: Multiple features based approach for automatic fake news detection on social networks using deep learning. Appl. Soft Comp. **100**(106983), (2021), ISSN 1568-4946. https://doi.org/10.1016/j.asoc.2020.106983.
34. González Ramírez, P.L., Lloret, J., Tomás, J., Hurtado, M.: IoT-networks group-based model that uses AI for workgroup allocation. Comput. Netw. **186**, 107745 (2021). https://doi.org/10.1016/j.comnet.2020.107745.
35. Ramirez, P.L.G., Taha, M., Lloret, J., Tomas, J.: An Intelligent Algorithm for Resource Sharing and Self-Management of Wireless-IoT-Gateway. IEEE Access. **8**, 3159–3170 (2020). https://doi.org/10.1109/ACCESS.2019.2960508
36. Kurbiel, T., Khaleghian, S.: Training of deep neural networks based on distance measures using RMSProp. (2017). https://arxiv.org/abs/1708.01911
37. LeCun Y.A., Bottou L., Orr G.B., Müller K.R.: "Efficient BackProp", Montavon G., Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 7700, Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_3.
38. ROBOT LEARNING.In: Jonathan, H., Connell Mahadevan, S., Kluwer, Boston (eds.) 1993/1997, xii 240 pp., ISBN 0-7923-9365-1. Robotica. **17**(2), 229-235 (1999). https://doi.org/10.1017/S0263574799271172
39. Qi, X., Luo, Y., Wu, G., Boriboonsomsin, K., Barth, M.J.: Deep reinforcement learning-based vehicle energy efficiency autonomous learning system. 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA. 1228-1233 (2017). https://doi.org/10.1109/IVS.2017.7995880.
40. Xiaojun, C., Feiping, N., Sen, W., Yi, Y., Xiaofang, Z., Chengqi, Z.: Compound rank-k projections for bilinear analysis. IEEE Trans. Neural Netw. Learn. Syst. **27**(7), 1502–1513 (2016). https://doi.org/10.1109/tnnls.2015.2441735
41. Yuan, D., Chang, X., Huang, P.-Y., Liu, Q., He, Z.: Self-supervised deep correlation tracking. IEEE Trans. Image Process. **30**, 976–985 (2021). https://doi.org/10.1109/TIP.2020.3037518
42. da Cruz, M.A.A., Rodrigues, J.J.P.C, Lorenz, P., Korotaev, V., de Albuquerque, V.H.C.: In.IoT – a new middleware for internet of things. IEEE Internet Things J. **8**(10), 7902-7911 (2021). https://doi.org/10.1109/JIOT.2020.3041699.
43. Zhou, L., Rodrigues, J.J.P.C., Wang, H., Martini, M., Leung, V.C.M.: 5G multimedia communications: theory, technology, and application. IEEE Multimedia. **26**(1), 8-9 (2019). https://doi.org/10.1109/MMUL.2018.2875256.

Albert Rego was born in Valencia, Spain on December 13, 1991. He received his bachelor's degrees in Computer Science and in Telecommunications Technology Engineering in 2015. In 2016, he received a Master Degree in Telecommunications by the Polytechnic University of Valencia. He received his Ph. D in Telecommunications in 2020 under the FPU national scholarship for his work on software defined networks. His lines of investigation are communications systems, multimedia and artificial intelligence. He is author of several papers and has cooperated in some international conferences, both by reviewing papers and being a part of a committee .

Pedro Luis González Ramírez (pgonzalezr1@ucentral.edu.co) is Lecturer of Department of Electronic Engineer, Universidad Central (Colombia). Currently Ph.D. student in Telecommunication at Polytechnic University of Valencia (Spain) and MSc in Pontificia Universidad Javeriana (Colombia). He has special interest in android application development and research in Telecommunications Network Engineering, IoT & IoE, Network Architecture, Protocols and Algorithms. He revised many papers as a reviewer and session's chairs in many international conferences. He also is a member of the Research Group Maxwell of Universidad Central and Junior Researcher in Colciencias Colombia.

Jose M. Jimenez received his Degree on Computer Science Engineering in 1997 and his University Masters Degree on Corporate Networks and System Integration and University Masters Degree of Digital Postproduction from the Polytechnic University of Valencia (Spain). He was Professor since 2005 for that University and since 2011 in the Technical School of Telecommunications Engineering. He is a Cisco CCNP Instructor since 2004. He worked as a network designer and administrator in several enterprises. He has several scientific papers in international conferences, international journals with JCR. He has been involved in several programme committees and in organisation of international conferences. His research interests include wireless sensor networks, ad-hoc and 2P networks, cloud computing, network security. He has worked in the private sector for 18 years focused on design and management of corporative networks .

Jaime Lloret (jlloret@dcom.upv.es) received his B.Sc.+M.Sc. in Physics in 1997, his B.Sc.+M.Sc. in electronic Engineering in 2003 and his Ph.D. in telecommunication engineering (Dr. Ing.) in 2006. He is a Cisco Certified Network Professional Instructor. He worked as a network designer and administrator in several enterprises. He is currently Associate Professor in the Polytechnic University of Valencia. He is the Chair of the Integrated Management Coastal Research Institute (IGIC) since January 2017and he is the head of the "Active and collaborative techniques and use of technologic resources in the education (EITACURTE)" Innovation Group. He is the director of the University Diploma "Redes y Comunicaciones de Ordenadores" and he has been the director of the University Master "Digital Post Production" for the term 2012-2016. He was Vice-chair for the Europe/Africa Region of Cognitive Networks Technical Committee (IEEE Communications Society) for the term 2010-2012 and Vice-chair of the Internet Technical Committee (IEEE Communications Society and Internet society) for the term 2011-2013. He has been Internet Technical Committee chair (IEEE Communications Society and Internet society) for the term 2013-2015. He has authored 14 books and has more than 600 research papers published in national and international conferences, international journals (more than 300 with ISI Thomson JCR). He has been the co-editor of 54 conference proceedings and guest editor of several international books and journals. He is editor-in-chief of the "Ad Hoc and Sensor Wireless Networks" (with ISI Thomson Impact Factor), the international journal "Networks Protocols and Algorithms", and the International Journal of Multimedia Communications. Moreover, he is Associate editor of "Sensors" in the Section sensor Networks, he is advisory board member of the "International Journal of Distributed Sensor Networks" (both with ISI Thomson Impact factor), and he is IARIA Journals Board Chair (8 Journals). Furthermore, he is (or has been) associate editor of 46 international journals (16 of them with ISI Thomson Impact Factor). He has been involved in more than 500 Program committees of international conferences, and more than 160 organization and steering committees. He has led many local, regional, national and European projects. He was the chair of the Working Group of the Standard IEEE 1907.1 from 2013 to 2018. Since 2016 till today he is the Spanish researcher with highest h-index in the TELECOMMUNICATIONS journal list according to Clarivate Analytics Ranking. He has been general chair (or co-chair) of 63 International workshops and conferences. He is IEEE Senior, ACM Senior and IARIA Fellow .