

Article

Deployment of IoT Edge and Fog Computing Technologies to Develop Smart Building Services

Francisco-Javier Ferrández-Pastor ^{1,†} , Higinio Mora ^{1,†} , Antonio Jimeno-Morenilla ^{1,*,†} 
and Bruno Volckaert ^{2,†} 

¹ Department of Computer Technology and Computation, University of Alicante, 03690 Alicante, Spain; fferran@dtic.ua.es (F.J.F.-P.); hmora@ua.es (H.M.)

² Internet Technology and Data Science Lab (IDLAB), Ghent University, 9000 Gent imec, Belgium; Bruno.Volckaert@ugent.be

* Correspondence: jimeno@dtic.ua.es; Tel.: +34-96-590-3400

† These authors contributed equally to this work.

Received: 26 September 2018; Accepted: 20 October 2018; Published: 23 October 2018



Abstract: Advances in embedded systems, based on System-on-a-Chip (SoC) architectures, have enabled the development of many commercial devices that are powerful enough to run operating systems and complex algorithms. These devices integrate a set of different sensors with connectivity, computing capacities and cost reduction. In this context, the Internet of Things (IoT) potential increases and introduces other development possibilities: “Things” can now increase computation near the source of the data; consequently, different IoT services can be deployed on local systems. This paradigm is known as “edge computing” and it integrates IoT technologies and cloud computing systems. Edge computing reduces the communications’ bandwidth needed between sensors and the central data centre. Management of sensors, actuators, embedded devices and other resources that may not be continuously connected to a network (such as smartphones) are required for this method. This trend is very attractive for smart building designs, where different subsystems (energy, climate control, security, comfort, user services, maintenance, and operating costs) must be integrated to develop intelligent facilities. In this work, a method to design smart services based on the edge computing paradigm is analysed and proposed. This novel approach overcomes some drawbacks of existing designs related to interoperability and scalability of services. An experimental architecture based on embedded devices is described. Energy management, security system, climate control and information services are the subsystems on which new smart facilities are implemented.

Keywords: fog computing; cloud computing; Internet of things; smart building; smart energy management

1. Introduction

Building automation systems use open communication standards and interfaces that can integrate a wide choice of different building control disciplines such as heating, ventilation, air conditioning, lights and blinds, safety features, and equipment. Pre-existing buildings, however, do not usually have these systems. Generally, each type of installation offers specific services: Heating Ventilation and Air Conditioning (HVAC) controls climate services, cameras and sensors offer security services, etc. Only when an energy management system is designed, the different subsystems are related, but only connected through the building’s energy management system. The energy management services are centralised in specialised software. Different manufacturers with different services find it difficult to integrate new services and functionalities, for both users and maintenance technicians. Automated buildings integrate software for control and data acquisition with industrial protocols and

interfaces. In addition, integrating new services into this type of solution is not easy. It depends on the installed software's development. These industrial developments also offer cloud-connected solutions and intelligent services for energy management. These services are also developed in centralised computer systems. The data are transmitted to these systems or to the cloud, where they are analysed. This paper proposes to use the edge and fog computing paradigms deployed in Internet of Things (IoT) technologies with two main aims:

1. To facilitate the integration of new intelligent and interoperable services in automated and non-automated buildings (*integration*).
2. To allow the distribution of smart services between all of the building's subsystems (*interoperability*).

Through this proposal, the relationship between a building's subsystems is facilitated. It also promotes the creation of new smart services (e.g., new distributed smart control algorithms, using data captured by power management to detect human activity, capturing pattern recognition of device connection, calculating renewable power prediction, using power data in security services, etc.). In this work, we designed a middleware architecture with two main layers based on embedded devices, IoT communication protocols and hardware support to develop artificial intelligence algorithms (Figure 1). To achieve this, we added two conceptual levels to the building's facilities: edge and fog node. Each level has different kinds of devices and functionalities. We deployed and implemented a layer-based middleware architecture to experiment the model.

This paper is organised as follows: Section 2 reviews smart building technologies, IoT deployment in buildings and the edge computing paradigm. Section 3 proposes a method to deploy edge and fog computing paradigms in buildings (automated or not). Section 4 presents the experiments conducted. Finally, Section 5 describes conclusions and future works.

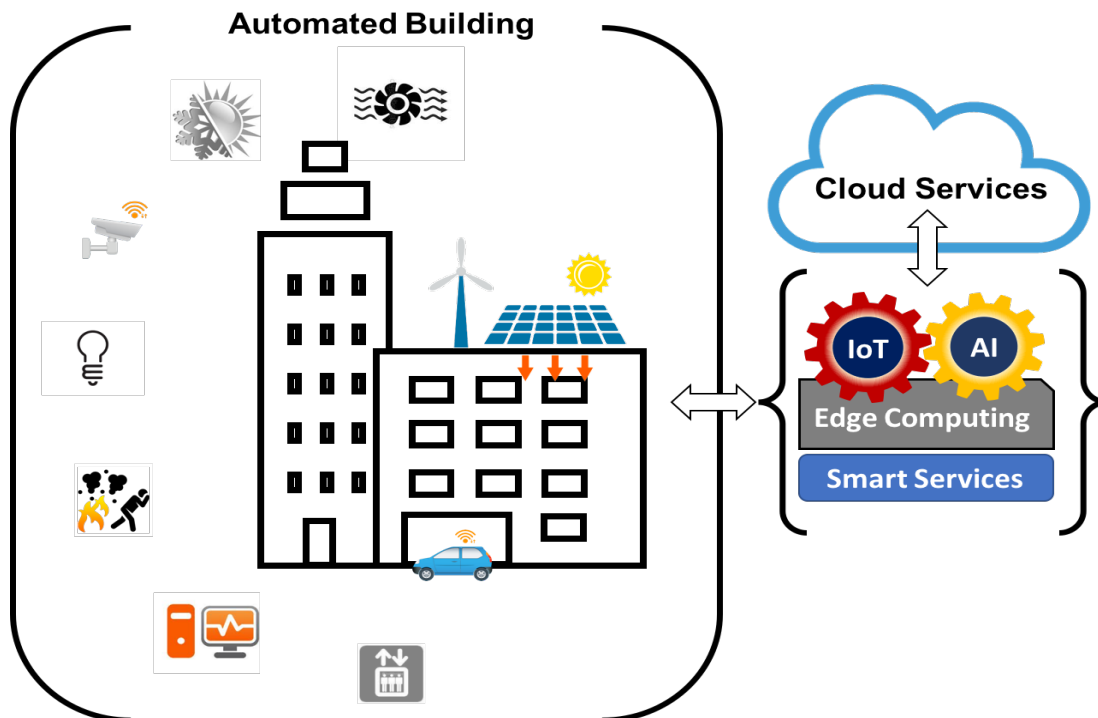


Figure 1. Automated building subsystems and information technologies context.

2. Related Work

This section addresses the main research lines related to this work. Firstly, we review the resource and service provisioning based on the edge computing paradigm, following by an analysis of the cloud

computing layer. Finally, we examine techniques for implementing smart buildings and summarise the contributions made by previous research in a final subsection.

2.1. Edge Computing Resource and Service Provisioning

The network was recently tagged at two ends called the “edge” and “core” ends to pinpoint where the processing takes place [1]. The edge end is close to the data sources and users, and the core end consists of the cloud servers. In this way, the edge computing paradigm [2] pushes computations to the edge of the IoT network [3], to reduce data-processing latencies and the number of data sent to the cloud. A cloud-based backend can serve processing requests that are less time-sensitive or of which results are not needed by the source devices themselves (e.g., big data analytics on the state of the IoT network).

In terms of edge computing resource provisioning, the ongoing Horizon 2020 RECAP project [4] proposes an integrated cloud–edge–fog architecture aimed at solving application placement, infrastructure management and capacity provisioning. Cloud/Edge infrastructure monitoring is enriched with application, infrastructure and workload models, which are in turn fed into an optimisation system that orchestrates applications and continuously configures the infrastructure.

The research conducted by Xu et al. [5] proposes a utility-aware resource allocation method for edge computing, called Zenith. With Zenith, service providers can establish resource sharing contracts with edge infrastructure providers allowing a latency-aware resource provisioning algorithm to schedule edge tasks in a manner that meets their latency requirements.

Edge Node Resource Management (ENORM) [6] is a framework that manages edge/fog node resources, enabling auto-scaling edge nodes by monitoring application needs. Specific applications can be prioritised through static priority-assignment. Provisioning and auto-scaling mechanisms are relatively simple implementations based on linear search.

The edge cloud paradigm is also viable when the sources themselves are mobile [7]. Intelligent computational offloading of mobile devices towards edge nodes, specifically at the edge of radio access networks, was studied by Chen et al. [8]. In this work, the authors provided task offloading algorithms formulating the distributed computation offloading decision as a multi-user computation offloading game. In the same line, Wang et al. [9] investigated the issue of jointly coordinating offloaded tasks to a multitude of edge nodes and propose introducing admission control at the edge level along with a two-phase scheduling approach, improving offloading performance compared to traditional closest-edge selection approaches.

2.2. Cloud Computing Service Provisioning

The cloud computing paradigm is one of the most innovative strategies in terms of the adoption of information technologies by society and industry [10]. The advantages provided improve efficiency and reduce costs, while providing on-demand IT resources and services that are ubiquitously accessible through the Internet.

Currently, there is a diverse range of cloud computing services [11] and even how they are offered [12]. It is an intensely researched topic where many works are being proposed; there are even review works that summarise cloud computing paradigm related studies [13,14].

This subsection describes the previous work on the following issues, which are related with this manuscript's topics: (i) security; (ii) Quality of Service (QoS); and (iii) edge-oriented service provision.

(i) Security is a challenging issue in cloud computing. Cloud services are located outside of the application environment and beyond the firewalls' protection, and, therefore, additional security layers are needed. In addition, the mobility and heterogeneity of the edge and fog computing applications make it difficult to define a single procedure. A distributed security strategy is therefore required. Furthermore, a standardised environment is necessary to properly address this issue and specify how the fog and edge devices can collaborate with each other [15]. The communication of sensitive data among multiple fog nodes on the edge of a network requires lightweight solutions for

resource-constrained things [16]. Another security-related issue is data location. It is common to run data analytics in the cloud. Therefore, the public vs. private clouds debate on data safety or privacy arises [17].

(ii) Resources allocated to cloud applications are generally set according to the contracted Service Level Agreements (SLA) [18]. However, in practice, bottlenecks can occur due to the saturation of the allocated infrastructure caused by occasional situations of massive transaction execution. To address this problem, the cloud infrastructure can be dynamically expanded when resources are available. Currently, the most innovative trends are aimed at building automatic SLA contract compliance systems. An extensive review of proposals related to quality service management can be found in the research conducted by Faniyi and Bahsoon [19], and Singh and Chana [20]. With this in mind, several strategies are proposed to anticipate applications' resource needs and QoS requirements. Recent works try to combine security and QoS issues to provide a comprehensive performance metric [21].

(iii) Finally, the misuse of cloud services is another issue in this area. An IoT environment is a dynamic execution context that the fog and edge devices are continuously joining or leaving. Elasticity in service provision at the network edge must therefore be provided [22]. To this end, sharing the application workload among the network's available devices can provide flexibility for advanced computing applications [23,24]. Reliable service provisioning methods are proposed to offer the system a higher resilience and provide agile and optimised cloud services [25]. In this topic, cloud frameworks and middleware technologies are set as a platform for communication with the cloud layer and among devices with different interfaces, operating systems, and architectures [26,27].

2.3. IoT in Building Services Engineering

IoT development offers new resources to develop digital services on buildings [28]. Common IoT applications in buildings are energy saving procedures, maintenance improvements, chore automation and security enhancements. Energy saving in buildings is an important topic due to global warming [29]. IoT can reduce greenhouse gases, not just locally but also on a grander scheme. At present, IoT is also used in the building sector to assist facility management [30]. IoT enables operational systems that deliver more accurate and useful information for improving operations and providing the best experiences for tenants. There are IoT-based proposals that show how building systems can communicate to the cloud and analyse the acquired data to develop new business insights capable of driving real added-value and greater performance [31,32]. Experimental studies show that IoT platforms can not only improve the interconnectivity of the entities in industrial energy management systems but also reduce the energy costs of industrial facilities [33]. FacilitiesNet [34] suggests that the Building Internet of Things (BIoT) is contributing to a significant shift in the way we acquire information, interact with each other and make decisions. The BIoT is not only about connectivity or the number of devices, but also about delivering real and relevant outcomes. Currently, there are many examples of smart home applications based on Internet of Things [35,36].

However, smart devices or "things" are just devices or embedded systems that are connected to a network [37]. The value-added comes from the ability of designing coordinated systems and providing intelligent services to offer real benefits. These features basically depend on the management of heterogeneity of different types of connected things and their interoperability, and on the potential of data processing to provide intelligence [38,39]. The research conducted by Tolga and Esra [40] concludes that IoT technology has not yet become stable in terms of both software and hardware in the smart home system. It is possible to show that one of the most important reasons for this is that the IoT technology is still in its developmental stage. The review about the future of edge-cloud and edge computing for IoT Applications composed by McElhannon [41] concludes similarly. This review mentions that the concept and development is currently in a relatively early stage and many challenges are ahead to be solved from both academic and industry perspectives.

IoT presents new opportunities, but many businesses are still looking to understand and analyse how it will impact and integrate with their existing IT structure and management strategies. To this end, specialised use models and technologies must be created to bridge this gap [42].

2.4. Findings

The following conclusions clarify the novel aspects of this work's research proposal:

- The general concept of cloud computing as a “utility” is well suited to the conventional needs of smart home applications. However, there are scenarios where moving all computations to the cloud is not realistic.
- Edge computing emerges as a computing paradigm to perform computations near to the data generated by the IoT devices. This paradigm might help to meet recent applications' security and QoS requirements.
- Currently, advanced building facilities that control subsystems usually use Internet, IoT protocols and web services. Proprietary systems are designed using standard Internet communication protocols for control and monitoring. Previous works show that the control systems for climate control, power management or security use different monitoring and control technologies, based on wireless sensor networks, web interfaces and industrial control models. Monitoring applications analyse these subsystems in supervisory and data acquisition systems. There are different solutions for different subsystems. Considering the scenario described above, the model proposed in this work introduces the following novel elements:
 1. A layered architecture (integrating both the edge and fog levels) and a method to provide interoperability between the subsystems, and to develop smart services in building control is introduced. The method uses edge and fog paradigms that integrate IoT protocols and operate AI techniques in a local Intranet. A communication layer with cloud services completes this layer architecture.
 2. A method based on user centred methodology to design, validate and improve new services under interoperability requirements is introduced.
 3. The proposal enables using non-proprietary hardware and software systems that can be implemented in buildings already built.

3. Computing Model Design

The facilities in the buildings are classified into subsystems (light, climate, energy, security, alarms, lifts, etc.). In automated buildings, these subsystems are controlled and monitored by specialised control technologies. In non-automated buildings, these services do not exist and the subsystems are controlled electronically and electrically. In both cases, all subsystems offer a necessary service in building operations. Logically, each subsystem acts in its scenario and is not designed to be interoperable with the other subsystems. Embedded electronic controllers and different connected sensors can automate each subsystem. These services are based on reactive control rules. In addition to embedded control systems and sensors, communication technologies (based on Internet protocols) and new mobile devices are offering new possibilities to develop control, monitoring and data access services.

Human interfaces and specialised applications, developed on smart mobile devices and connected to web servers, are examples of the services that have been implemented over recent years. Experts in each subsystem (climate, security, power, etc.) have knowledge that can be converted to expert rules. These rules are converted into control algorithms for management, maintenance, control, optimisation and other activities. These rules are programmed and implemented on programmable devices; however, they are static, do not evolve when new conditions occur and are neither interoperable nor adaptably to each installation's characteristics. For example, climate or security experts decide how to configure each subsystem using standard start conditions. Each control rule only works in one subsystem (climate or security in this example), and, therefore, there is no interoperability between

these subsystems. Considering this situation, the model proposed facilitates and allows the integration of new digital services based on the interoperability of the different subsystems and introducing new services based on artificial intelligence (AI) techniques to the current facilities. For example, facilities such as lift control can be used for a security service or a building energy management service. Climate control facilities may be integrated with security subsystem. A weather forecast software system integrated into the model may be used by energy management services or building air conditioning services.

The aim is that the experts in each subsystem participate in designing integrated services and convert all subsystems into an interoperable system. This model develops automatic rules and allows decision-making considering installation's behaviour itself. The model is based on a process consisting of four development phases (Figure 2) and a hardware–software architecture divided into different levels (Figure 3). The main levels of this architecture are the edge and fog levels. These two levels introduce the novelty of applying IoT technologies in buildings. The model's phases (analysis, design, implementation and start up) are introduced below.

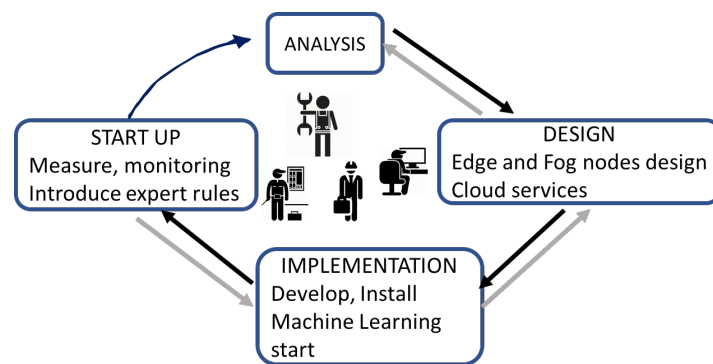


Figure 2. Model based on user centred relations.

- Analysis: Different expert users are identified in this phase (climate, security, power, water, energy, managers and Information and Communications Technology (ICT) technicians). Expert users are consulted to specify the main processes that need controlling. ICT experts participated in this process as a link for integration. This first approach produced the *things* (objects) required to design control rules and potential services. In this phase, user-centred methodology is used and the subsystems requirements are captured.
- Design: We propose a three-level architecture (edge, fog and cloud), as shown in Figure 3.
- Implementation and data analysis: The subsystems are installed and integrated during this phase. Services are based on rules in each subsystem. The data generated by the things (objects) are analysed to design machine-learning based services.
- Start up: Initially, expert rules are developed with supervision for each subsystem. Then, rules are installed with feedback processes. Finally, automatic and adapted rules are inferred through artificial intelligence techniques.

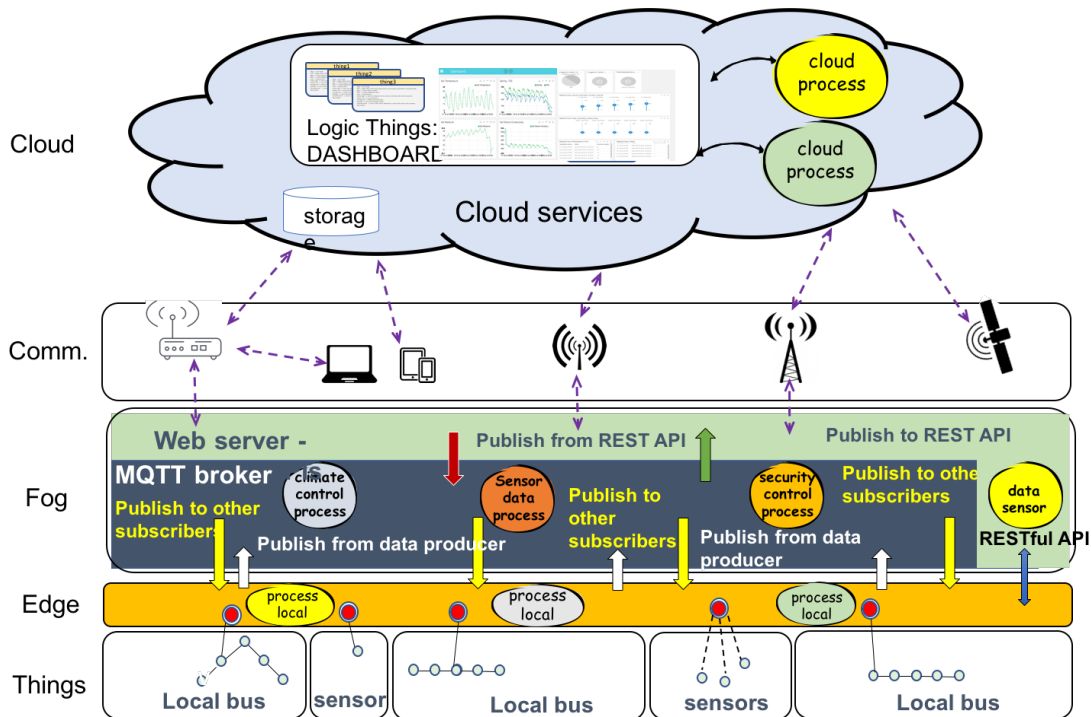


Figure 3. Communication architecture. Each level has a different functionality. Two communication levels are proposed: IoT (using Message Queuing Telemetry Transport (MQTT)) and Web (using REpresentational State Transfer (REST) protocols). These protocols' layers cover the established requirements of integration and interoperability.

3.1. Analysis and Design

Expert users undertake different reviews of this process. User-centred techniques are used to design integrated processes. The aim is to obtain all *things* (objects) required, their relationships and their potential services. Once the *things* (objects) and services are specified, the communication protocols and control technologies must be related. IoT protocols and embedded controllers are chosen. Human interfacing is proposed. Edge and fog layers with their functionalities are specified. Expert rules and intelligent services are analysed. Finally, maintenance and operation methods are proposed. All of these tasks are shared between an expert technician and information technologies expert.

The results are the definition of the *things*, their relationship and the interaction with the edge and fog layers. All of the building's subsystems are represented in this process. Data sensors, actuators, controllers, rules and processes are designed allowing the integration of all subsystems. The dataset, objects and devices are represented by the *things* concept. A *thing* is formed by an entity and a context with state and configuration data. *Things* data are in different configuration files stored in fog and edge nodes.

A *thing* is represented in a data vector: $[ID, type, node, context]$.

- *ID* is an identification code.
- *Type* can be a sensor, actuator, variable, process, device, interface, data storage or any object that can write, process, communicate, store or read data in the IoT ecosystem.
- *Node* specifies the building subsystem, functionality description, layer type (edge, fog, communication or cloud), IoT protocol and time access.
- *Context* represents the time, date, location, relations with others things, the state and the access rate that use to publish or read data in the IoT ecosystem.

Table 1 is an example of this data structure formed by *things* ($[ID, type, node]$). All *things* can access to configurations files (CF) to know how to use the available data and to configure new data

with the appropriate access permissions. Context data are in memory or static storage. Different control rules are designed using the *things* defined. These control rules are part of control processes that are distributed in different embedded systems connected to the network. The *things* represent all available resources deployed in the different subsystems of the installation. At this level, all *things* are analysed, specified and related by designers. Basic control algorithms are implemented using this information. The configuration files allow the interoperability of all *things* between layers and devices.

Table 1. Thing example description. The ID, Type and Node data are written to configure XML file. The configuration files are stored in fog nodes.

Thing ID	SCP1R1
Type	Temperature sensor
Node	[climate subsystem, edge, IoT protocol (MQTT, HTTP.), time access]
Context	[time, date, location, other things relations, state]

Another level of design, in this phase, must propose node requirements and specifications used in the *things* management. Processes and services designed will be implemented into edge or fog nodes. Each node must be specified to determine its internal functions, communication and its services. Intelligence and processing capabilities are developed where the data are obtained. Edge and fog layers have nodes that are near the data sensor, actuators and controllers. The method proposed in this paper uses two layers (edge and fog) with two functionalities. Each layer can deploy a network of interconnected nodes that facilitate interoperability.

The functionalities of edge and fog layers are:

- Edge layer functionality: Control software developed on embedded devices that connect sensors/actuators. Some AI algorithms can be installed on an edge node. Central Processing Units (CPU) and computing resources are limited. Communication interfaces are installed to allow integration in the local network.
- Fog layer functionality: Communication, AI paradigms, storage, configuration files, and monitoring activity on the local area network level. A fog node processes data in an IoT gateway, server device or other device with processing, communication and storage. Local, global and integrated services are implemented at this level. Algorithms based on machine-learning paradigms are developed using the hardware, software and communication capabilities of these nodes. Fog layer device can also perform the edge node functions in facilities or services with few units.

With these two levels building facilities can be optimised to obtain integration and interoperability between the different subsystems.

Table 1 shows how each *thing* is related to a context and to a node specification. The node identifies the subsystem to which it belongs (control, energy, climate, etc.), the layer (fog, edge, communication, and cloud) and the functions it performs.

3.2. Architecture Design

Objects (things) and their relationship are obtained in the analysis and design phase. Specifications and requirements are used to implement each layer. Implementation depends on the design and existing technologies (hardware, communication, and software) that provide the required functionalities. An architecture adapted to the existing facility is developed in this phase. IoT protocols provide interoperability and AI paradigms offer adaptability and optimisation. Edge computing nodes are used on control devices and fog computing nodes are installed on local network nodes. These levels provide powerful resources to configure, install and operate new processes.

IoT protocols communicate all subsystems data. Each subsystem is composed of objects/things (virtual level), installed as sensors/actuators/controllers devices (hardware level) that can be connected.

The requirements established in IoT communications for building scenarios are: standard protocols, low-power consumption, easy access and maintenance, support to integrate new modules, non-proprietary hardware or software and low cost devices. One of the main IoT protocols, designed to work on heterogeneous communication scenarios that provide resources to integration and interoperability, is the MQTT protocol. This protocol is proposed as communication paradigm between sensors, actuators, controllers, communication devices and subsystems.

Some of the MQTT protocol's main features are shown in different works [43–48] which makes it especially suitable for this research. Some of them are:

- It is a publish–subscribe messaging protocol developed to resource-constrained scenarios.
- It has low bandwidth requirements.
- It is a very energy efficient protocol.
- The programming resources are very simple making it especially suitable for embedded devices.
- With three QoS levels, it provides reliable and secure communications.

MQTT develops ubiquitous networks that allow an *n-to-m* node communication model. Any node is able to query and be queried by other nodes. In these scenarios, any node may play the role of a base station, capable of transmitting its information to remote processing places. The nodes, in a Ubiquitous Sensor Network (USN), can process local data; if a gateway is used, they have global accessibility; and they can offer extended services.

A node (edge or fog) can have local and global access. These facilities have different possibilities and benefits. Local data processing is necessary for basic process control and global processing can be used for pattern detection and information generation. In this sense, the proposed platform uses a combined functionality: different USN over a local network area connected to cloud-IoT services. In this scenario, a computing layer (edge or fog level) will serve as an interface between control processes and cloud-services. This layer can process data before communicating to the cloud.

Three actions are needed to implement edge and fog computing nodes:

- Connection and communication services: All devices must be in the same network and be interoperable. All sensors and actuators can be accessible to develop services. An example of this activity is the remote reading of building's power parameters, ambient conditions and open weather forecast data on the Internet. Other functionalities such as security, reliability and interoperability of connections should be implemented in this activity.
- Control algorithms and data treatment in embedded devices (edge computing layer): In this activity, basic control rules and data analysis services implemented in these devices can develop new capabilities. This phase can be applied to data filtering, calculating climatic data or analysing power consumption, reactive control or detecting events using pattern recognition techniques.
- Advanced services on gateways nodes (fog computing layer): This level uses and manages AI paradigms and IoT communication protocols. Fog computing nodes perform an intelligent analysis of the data, stores it, filters it and communicates it to the different levels, either to correct new control actions at lower levels, or produce information of interest to services in the cloud. Example of applications in this phase are analysing new patterns, predicting water or power consumption, smart detection and other predictive services.

Figure 4 shows an example of how fog and edge levels are integrated when building facilities are already installed. The possibilities of using this model also includes said buildings. When building subsystems are already installed, it is necessary to interleave embedded devices (edge-nodes) between controllers and sensors/actuators. Figure 5 shows the experimental scenario where the model is applied.

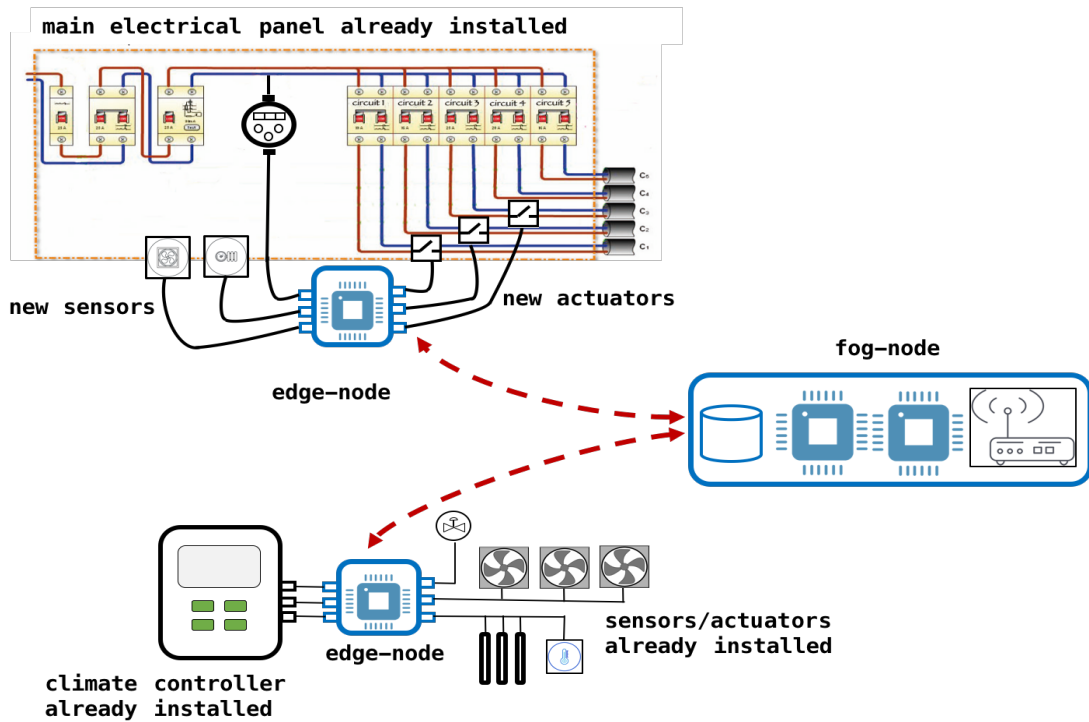


Figure 4. Example of edge-fog architecture implemented on buildings’ existing facilities: edge nodes are the lower level. New connections with the devices installed must be performed. Fog nodes that interconnect all subsystems are implemented integrating new devices connected to edge nodes. Edge and fog nodes can be deployed in all building subsystems.

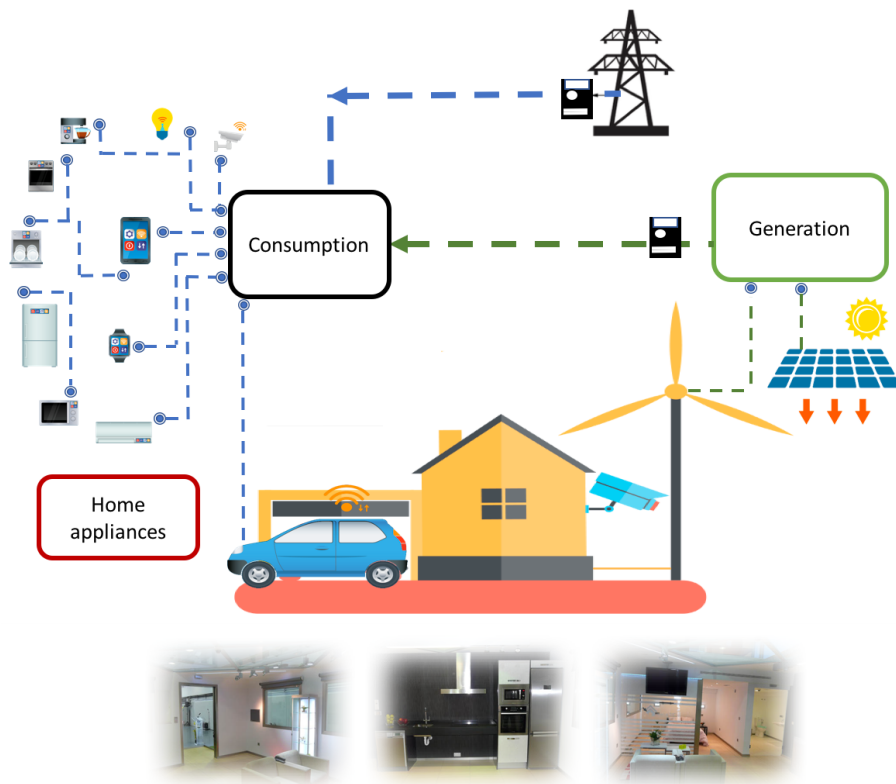


Figure 5. First experimental work in a residential building.

3.3. Test and Feedback

Standard methods are used during the test phase. Edge and fog layers offer different functionalities. Machine learning models are proposed for different subsystems and can be installed on edge or fog nodes. The following must be carried out to test machine-learning applications:

1. Define and capture datasets: Main variables must be identified, captured and stored. In different building subsystems, process datasets are data captured by sensors connected to the edge-layer. The datasets are monitored and stored using communication protocols. An example is a power metre connected in an electric panel to an embedded device (edge-node) that communicates power data to store and process in a fog-node device.
2. Training dataset and pattern recognition model. A subset of the previous dataset is used to train different models. Evaluation tests the model against data that have never been used for training. The results of this process are validated with expert users. The aim is to obtain a set of representative results to know how the model might perform in the real world.
3. Validation in a real scenario: New services and control algorithms must be implemented on edge and fog nodes. The models have algorithms that analyse the data, implement specific patterns, and use the results to develop optimal parameters. In this phase, the model can be modified or refined.
4. Test results in statistical terms and model evolution: Models based on AI algorithms will produce approximations and not exact results. Applications results are analysed to determine the level of confidence and allow model evolution. This activity supports developing new AI services or modifications on the algorithms implemented. Supervised and automatic changes are processes to maintain and improve the system. The processes in this phase include all model layers.

These activities are proposed for any installation that uses the edge and fog levels. As indicated previously, this model can be installed both in pre-existing buildings and new builds. For new building designs, installation based on the proposed model is easier to integrate. Moreover, the potential of the services that can be offered also makes it attractive for pre-existing buildings.

4. Implementation of Smart Services in Building Subsystems

The model is tested on a pre-existing residential building. Power management, control and monitoring services are designed and implemented. IoT protocols (MQTT and HTTP) and ML paradigms are used on the proposed layer architecture. KNN based Machine Learning approach and tree decision algorithms are used to manage power consumption (home appliances) and renewable generation (wind and solar). Monitoring and statistical data are implemented on cloud platform using a fog node in the house. This node is connected to different edge nodes that control the renewable and home appliances subsystems.

In Figure 6, an edge node is integrated in a previously installed renewable subsystem. With this new device on the edge layer, power management, security control and operating processes are integrated and become interoperable with other subsystems. New smart services can be designed. The edge node communicates data to the fog node gateway that manages power consumption and generation, and controls home appliances. Inputs in this node are data of renewable energy generation. Output controls are the ON-OFF switches to optimise generation, security and operation.

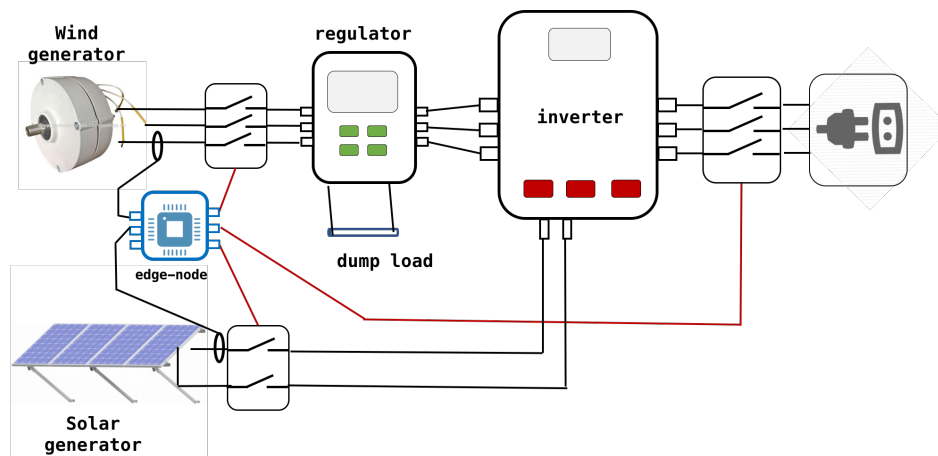


Figure 6. Example of an edge node integrated in a previously installed renewable subsystem. This node can control ON–OFF switches with new algorithms to manage the generation processes and communicate and monitor power data.

4.1. Analysis and Design

A residential building is analysed to design power management, security and control services. In a first approach, the main *things* (objects) required, their relationships and different services are shown in Table 2.

Table 2. Analysis and design requirements in experimental work.

	Analysis and Design Requirements	Things/Layer/Device/Process
renewable generation	Wind and solar generation are facilities that should be considered as necessary for every building. To this end, they are analysed for introduction to domestic buildings. The proposed things will be scalable to larger buildings. Generation resources are analysed to be integrated into the layers model.	things: power-meter, wind-solar forecast power, outdoor climate sensors, ON–OFF controllers, data consumption. edge-layer: control-communication algorithms, embedded devices fog-layer: subsystem integration and machine learning algorithms, communication and processing devices adapted comm-layer: local and cloud communication devices cloud-layer: dashboard data, storage and events
power consumption	All buildings have an electrical consumption that must be monitored to optimize their use. Consumption sensors obtain data that will be used to extract consumption patterns. With these consumption patterns, actions will be taken to reduce consumption and make better use of renewable generation. The control of loads (ON–OFF of household appliances) or people activity monitoring are services that are implemented for energy management, control and security in buildings.	things: indoor sensors (temperature, humidity, luminosity), ON–OFF climate controllers, energy meter. edge-layer: control-communication algorithms, embedded devices connected to air conditioning machinery. fog-layer: subsystem integration and machine learning algorithms, communication and processing devices adapted comm-layer: local and cloud communication devices cloud-layer: dashboard data, storage and events
home appliances control	Home appliances can be controlled by power management services. Actuators and sensors can also be used in security comfort and security services	things: ON–OFF home appliance controllers. edge-layer: control-communication algorithms, embedded devices connected to air conditioning machinery. fog-layer: subsystem integration and machine learning algorithms, communication and processing devices adapted comm-layer: local and cloud communication devices cloud-layer: dashboard data, storage and events
cloud services	Analytic processes, user interfaces and storage services must be designed. Tables and graphs with statistical data show data in real time. IoT communications store main data. Analysis applications generate information about the growing process	things: graphs, data tables, variables with things objects, events cloud-processes user interfaces, data storage, statistical calculations, analytics





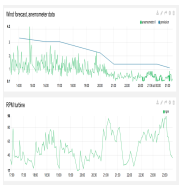
4.2. Implementation

Building subsystems in this house are analysed to integrate the model layers: edge control, fog services, communication to cloud and cloud services. Sensors, actuators and control processes (*things*) used in this experimental work are elected. The embedded devices used are shown in Table 3.

Response times and interoperability are required for control processes in home services. Human interfaces, data access and analytic services are services on local and cloud computing. Both of the above-mentioned requirements are treated with different protocols: MQTT on control/communication and HTTP (RESTful API) on cloud services are the IoT protocols to integrate and make interoperable all subsystems. In the layers model proposed, Machine-to-Machine (M2M) applications are also developed using the MQTT protocol, control, data process and data communication to cloud using RESTful protocol.

MQTT uses an open message protocol that enables the transfer of telemetry-style data (i.e., measurements collected in remote locations) in the form of messages from devices and sensors, along unreliable or constrained networks, to a server (BROKER). Messages are simple, compact binary packet payloads (compressed headers, much less verbose than HyperText Transfer Protocol (HTTP)), and fit simple push messaging scenarios such as temperature updates or mobile notifications well. Messages also work well for connecting constrained or smaller devices and sensors to a web service, for example.

Table 3. Embedded devices used in experimental house.

Features	
	<p>Fog node used: RaspberryPiARM BCM2837-system-on-chip (SoC)/CPU 1400 MHz memory 1 GB (OS Linux) OS multi threaded</p>
	<p>Edge node used: Broadcom BCM43362 Wi-Fi chip STM32F205RGY6 120 Mhz ARM Cortex M3 Real-time operating system (FreeRTOS) OS monoprocess</p>
	<p>ARM smartphone (Android/iOS/windows OS)</p>
	<p>Communication node: Quad band GSM: 850, 900, 1800, 1900 MHz GPRS class 12 data rates: DL max. 86 kbps, UL max. 86 kbps Internet services: TCP, UDP, HTTP, FTP, SMTP, POP3</p>
	<p>Cloud platform: Ubidots device API, connect any hardware to the Ubidots Cloud over HTTP, MQTT, TCP, UDP</p>

The MQTT communication protocol makes all objects interoperable. A publisher and subscriber model implemented by this protocol can interconnect all devices and things. This communication layer is managed by a broker device installed on a fog node. Different publishers and subscribers are implemented on different nodes. One gateway device (fog node) and two embedded controllers (edge nodes) are installed to control home appliances and power management. Things and processes are deployed on all nodes. Edge nodes control subsystems and the fog node implements AI paradigms

based on decision trees with rules defined by experts. Fog device communicates data to a cloud platform to develop dashboard screens for monitoring the subsystem's state. New cloud platform services can be developed: events detection, machine-learning processing, statistical analysis, etc. Expert users design basic control algorithms. After the learning and training process, these algorithms will be adjusted and modified following the expert system results. In this work, the aim is to optimise resources (control and energy) without losing productivity. Different control processes are executed in edge or fog nodes; the classification process and decision trees are implemented in a fog node. Algorithms are implemented in Python language. Open source libraries for this language are used for different applications.

4.3. Deploying and Testing

For pre-existing buildings, edge nodes are interleaved in installed controllers, electric panels, and sensors and actuators. Some new sensors/actuators are installed if new *things* are specified during the analysis phase (energy meter, climate, and controllers). The edge nodes deployed in this work have the following advantages:

- They do not interfere with the previous installation operation.
- They introduce new controls using new expert and automatic rules.
- They test and reconfigure expert rules designed in analysis, learning and testing validation.

Figure 7 shows the node (edge and fog) deployed on distribution panels. Power meter, ON–OFF switch controls and AI services are designed and installed in this node.

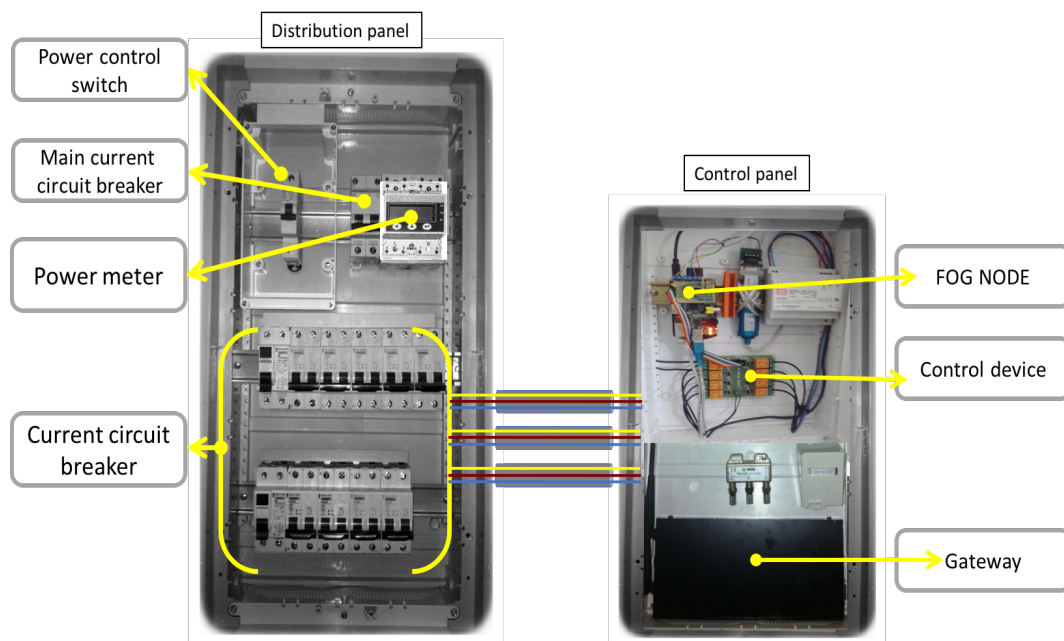


Figure 7. Nodes deployed in electric panels. Data capture, control algorithms, data analysis, storage and communication services are developed in different nodes using IoT protocols communication.

In the power management process, a time schedule with the selected flow rate is programmed by the expert user according to power consumption, power generation, consumption load curve, climate data and climate forecast data. The edge node captures the data and sends it to the fog node. The fog node processes diary data, indoor and outdoor ambient and weather conditions. Fog nodes can also capture other sensor data. These data consumption and generation patterns in the house are detected and classified. Consumption and generation results are added as data to analyse it together with the stored ones. Smart services as home appliances or human activity detection can be developed using Machine Learning methods (Figure 8).

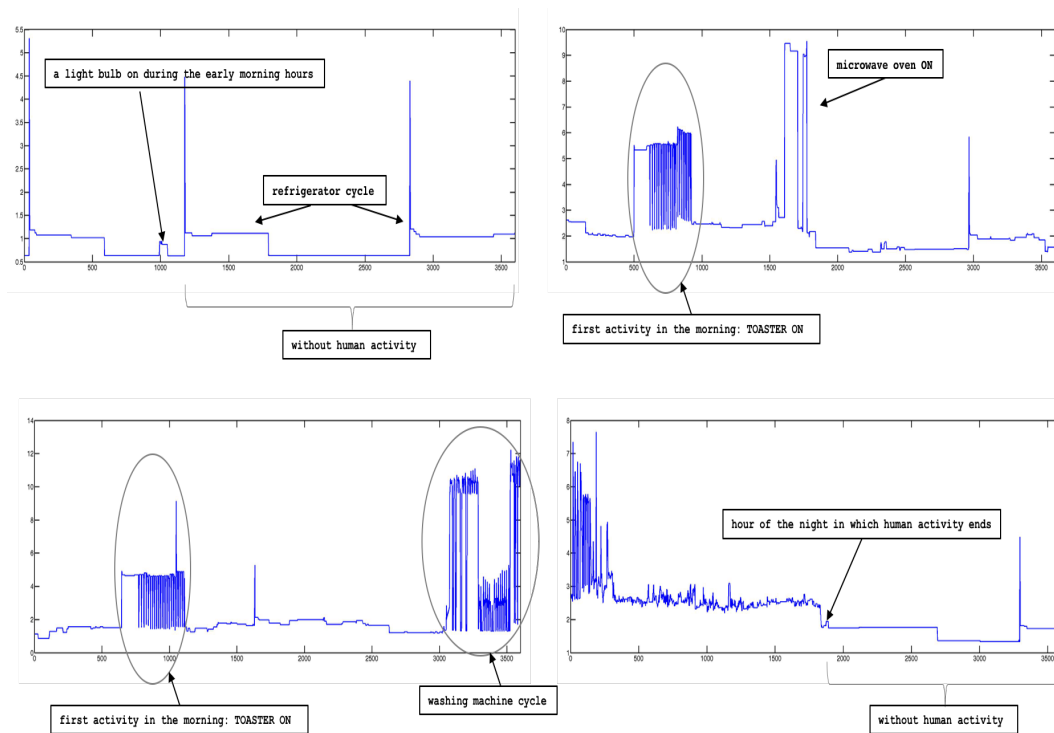


Figure 8. Deployed smart power functionalities. Classification process implemented in a fog node can be used to detect electric connections and human activities. Other services can be implemented using IoT communication.

4.3.1. Machine Learning: Data Capture Process (Edge-Node) and Home Appliances Classification (Fog-Node)

Power meter connected in the main distribution panel is used to capture data and to develop a pattern recognition model using the standard K-Nearest Neighbors (KNN) [49] classification algorithm. KNN is one of the most common methods in machine learning systems [50]. Power meter capture electric current; if a new home appliance is connected, the electric current data changes. Different home appliances have different change levels. Main variable used to identify different patterns of home appliances are the difference in levels of the electric current when they are connected. The *data capture process* flowchart (Figure 9) shows the algorithm implemented in an edge node to capture pre-process and communicate electric data.

In this process, a supervised phase uses a training dataset. Next, validation in real scenario tests the classification model. Home classification appliances will be used in different services: recognition of human activity, load control, renewable energy management, air conditioning, security, etc. In the training phase, different switch ON home appliances have been captured to obtain a set of patterns. Each household has a centroid vector that will be used for its detection in the classification process. The classifier process has as inputs the data of electric currents when connections are produced, as indicated in the algorithm shown above. The *KNN classification process* flowchart (Figure 10) depicts the KNN method. It is implemented in a fog node.

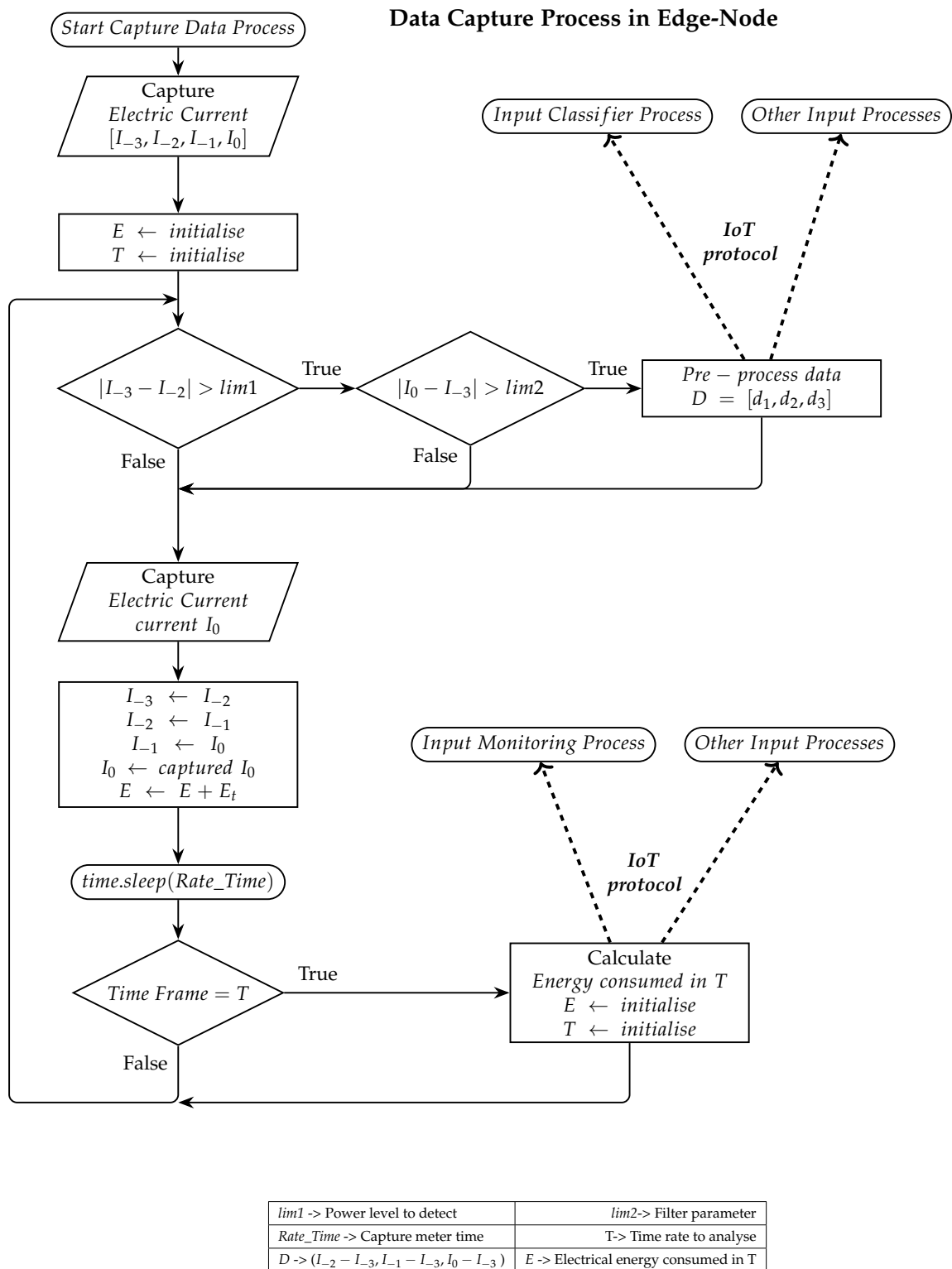


Figure 9. Captured and pre-processed electric consumption data in an edge node. The MQTT protocol is used to communicate data. In addition, the captured data can be used by other nodes to offer other smart services. Integration and interoperability is deployed.

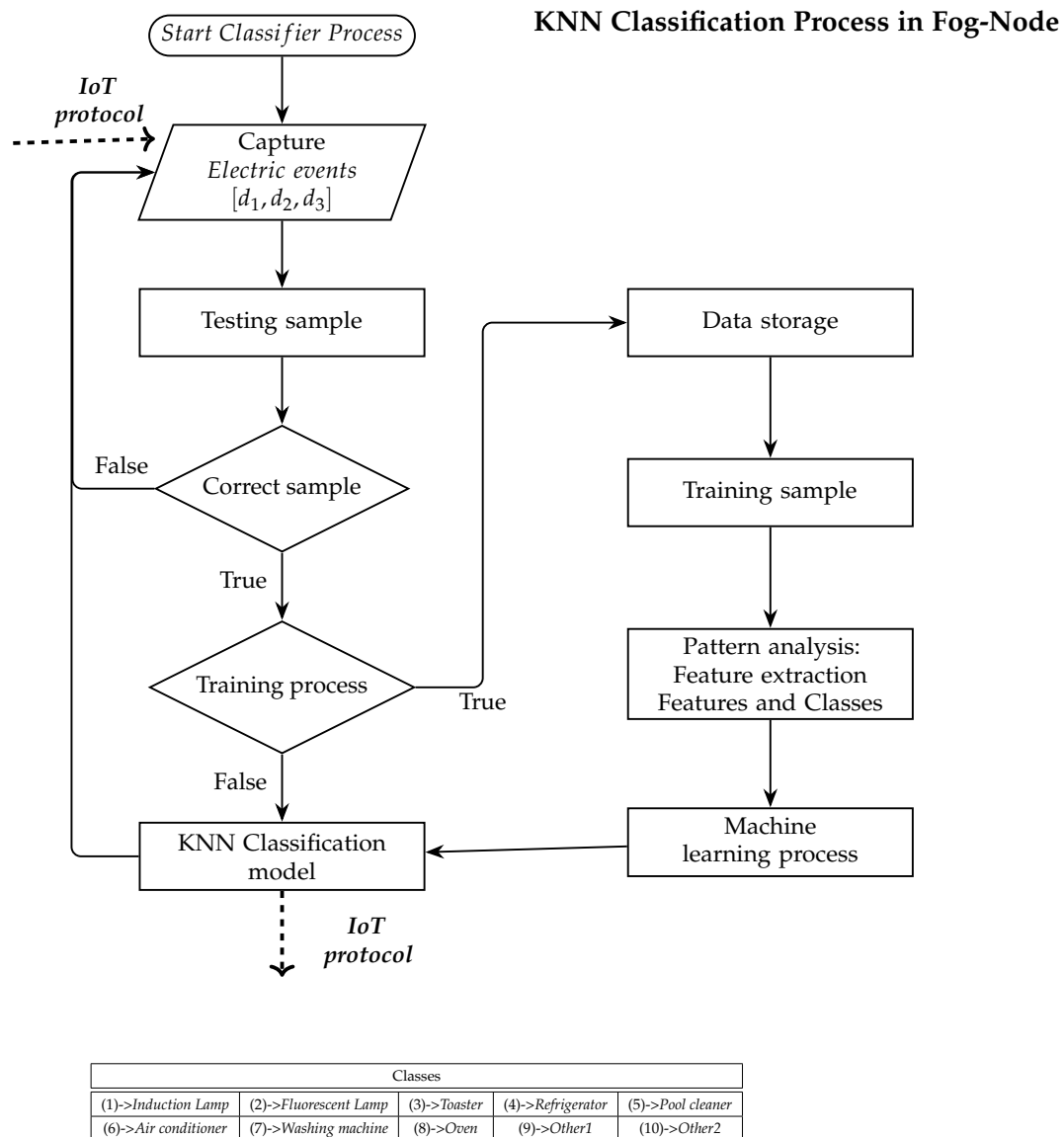


Figure 10. Classification process. Captured electric data are processed to detect home appliances connections. Other smart services can be designed using IoT protocol integration.

4.3.2. Renewable Power Management. Decision Tree to Control Electric Self-Consumption

Each building has a different demand curve, as well as a specific situation in terms of access to the grid. To this end, integration and interoperable facilities can implement different solutions that adapt to each situation, offering optimum management of solar–wind resources, optimising power efficiency, simplifying management processes and achieving the highest possible cost savings. In installations with AC coupling to the grid exists a problem when renewable energy exceeds the energy consumed. In experimental work, solar photovoltaic (PV) energy during the central hours of the day is greater than the energy consumed (Figure 11). However, after analysing the consumption curve, loads can be connected during this period of time to avoid injection into the grid. This requirement can be treated designing an algorithm that predicts when this event arises to connect different loads automatically. Using the integration and interoperable communication of all sensors and actuators, an algorithm implemented in different nodes has been developed (Figure 12).

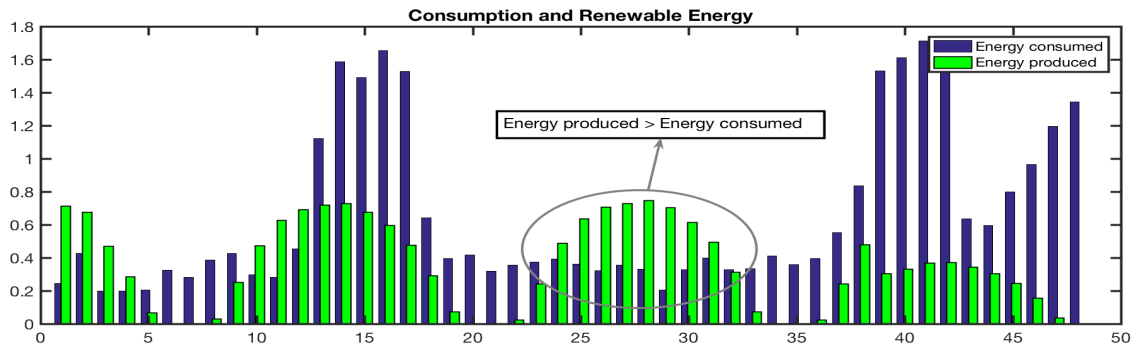


Figure 11. Figure shows consumption and production data in experimental work. In electric self-consumption facilities, without storage and without injection to the grid, the energy produced must be used in real time and must not exceed the energy consumed. The energy manager must predict this event and connect charges in advance.

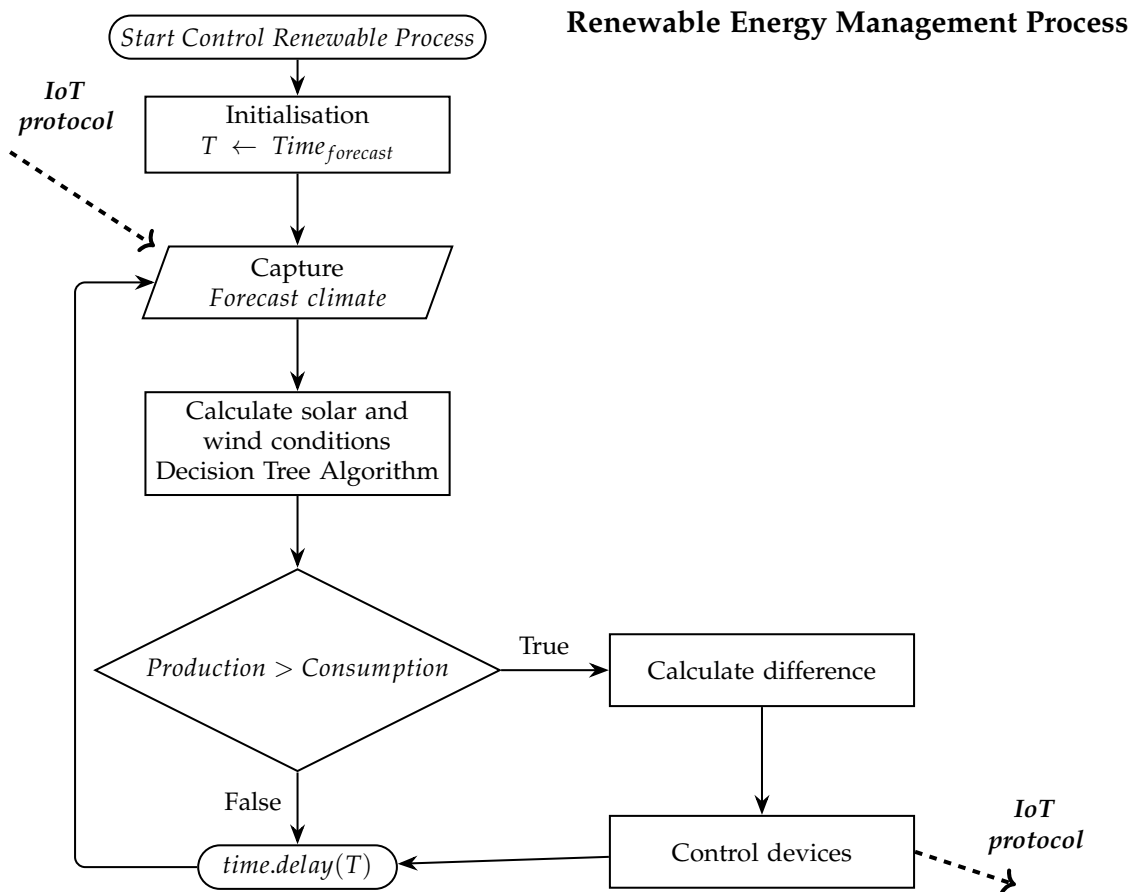


Figure 12. Renewable power management in electric self-consumption facility.

Figure 13 is an example of an algorithm developed in the power management subsystem. This process can be installed on an edge node. This node obtains the climate data forecast and predicts if the system can use renewable energy without storage.

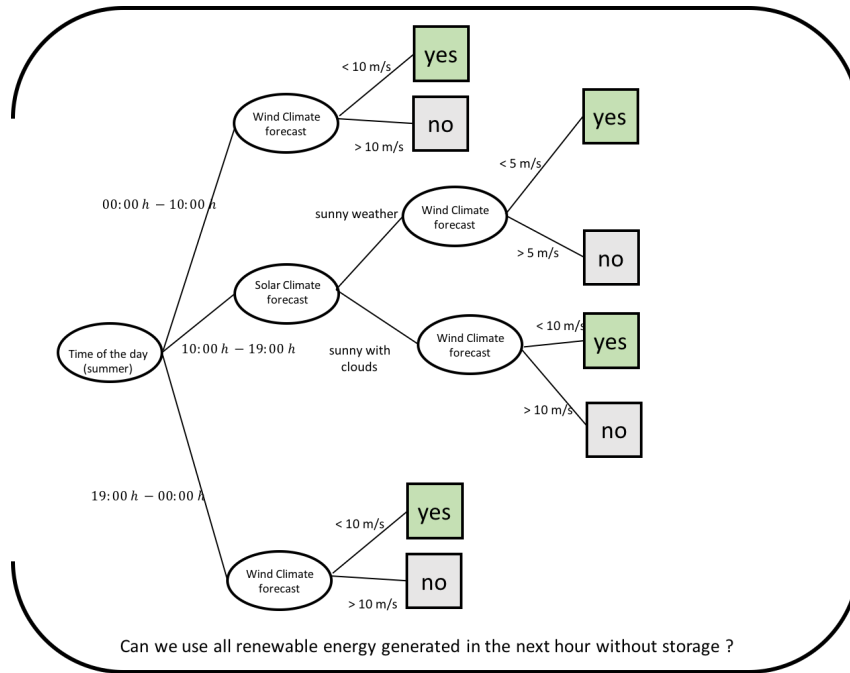


Figure 13. A decision tree developed on the power management subsystem. It is designed by an expert user and integrated on the edge node. This decision tree aims to optimise renewable energy use.

4.3.3. Control Home Based in Edge and Fog Nodes

Figure 14 shows an edge node installed in a room of a residential home. This node can control four devices (appliances) and capture sensor data (power consumption, generation, temperature, humidity, etc.). This device can communicate using the MQTT protocol. This protocol allows other types of communication between devices: smartphones, new edge nodes, etc. Figures 7 and 14 show a standard implementation that can be deployed in other buildings. In all systems, there are electric panels that deploy fog nodes and edge nodes, as shown in the figures.

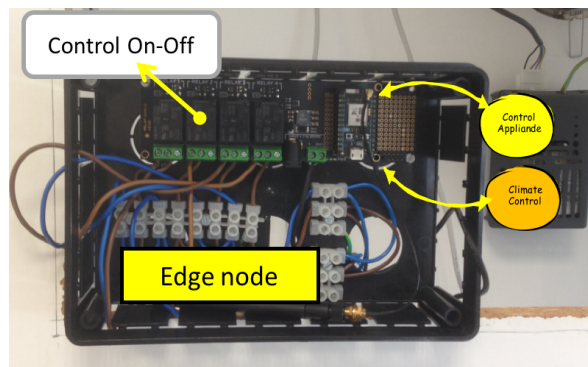


Figure 14. Edge node deployed. This node can control ON–OFF switches with new algorithms, and communicate and monitor sensor data in in each room or building stay.

4.3.4. Cloud Services Using IoT Protocols

Cloud services can monitor data accessed through fog nodes or a Human–Machine Interface (HMI). IoT protocol (MQTT) push data from any connected Internet-Enabled Device. Other services such as events detection, storage or statistical analysis complete this resource’s functionalities. Different platforms with similar services show the state of commercial IoT technology: Amazon IoT [51], Microsoft Azure [52], Ubidots [53] and Thingspeak [54], are some examples of companies that provide

IoT platform. Resources and an Application Programming Interface (API) communication between clients and the IoT platform are offered so that they can be used. HMI resources to design dashboard monitoring and control are one of the main utilities on these platforms. Fog nodes communicate data and information using cloud API. Other control services can be implemented. Dashboard designs to monitor data are pre-built on these cloud platform. Using API utilities, processes in fog node send data to each dashboard. The API documentation specifies the structure of the data that are exchanged between devices, the IoT platform and Mobile-Alerts Cloud, along with code examples and libraries to speed up the project. Figure 15 shows a dashboard designed on the Ubidots cloud platform. Ubidots is the IoT platform used in this experimental work. The model can use different standard protocols in the layers and different platforms that implement these protocols. Figure 16 shows configuration of events IF *variable* THEN *actions* in cloud platform. This functionality is offered by most IoT platforms.

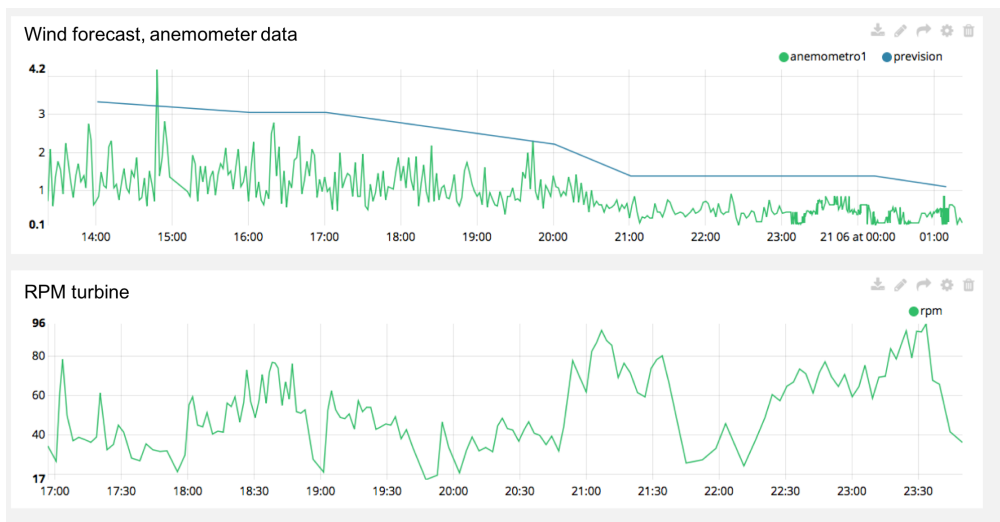


Figure 15. Dashboard configured on the cloud platform. Power wind generation data and forecast wind are shown.

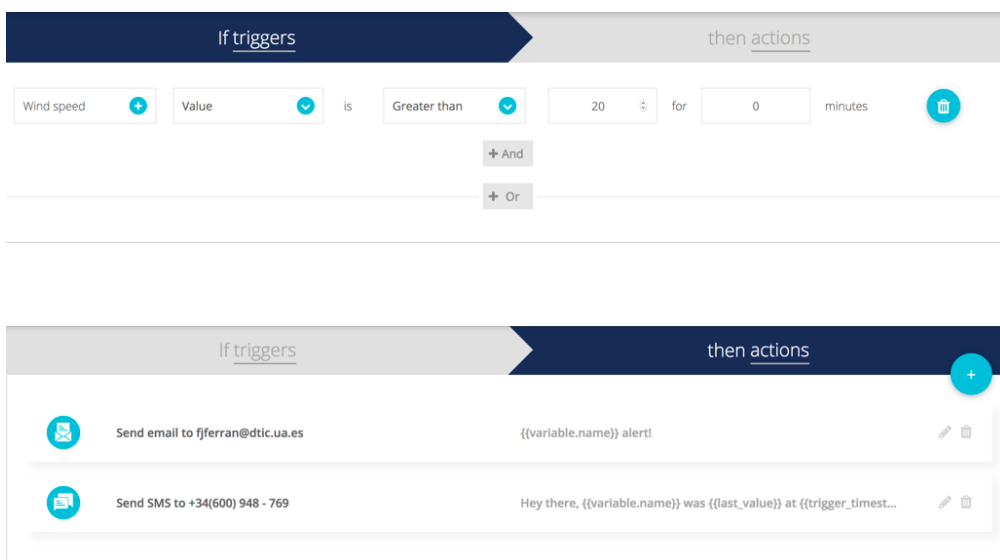


Figure 16. Dashboard on the cloud platform to configure events: IF event THEN action. This service shows how a facility can be controlled using the cloud access. Internet communication with fog nodes can control different subsystems in buildings and communicate events using email, SMS or other Internet services.

5. Conclusions

Edge and fog paradigms are being increasingly proposed for designing IoT systems. However, each paradigm offers solutions in a specific field of application. Integration and interoperability between different subsystems can improve this situation and offer better services. The main aim of this work is to address this issue by proposing a computing model based on a two-level architecture: edge and fog layers. These two layers allow designing and developing new services based on new kinds of useful information produced with data captured using embedded devices in edge or fog nodes. These nodes use a cloud platform and IoT protocols such as MQTT. MQTT is proposed and experimented as a communication protocol between different layers (fog–edge–cloud). The cloud platform is used to develop dashboard information panels and new services on the Internet such as control, storage and communication events. This platform can be used to deliver different services by means an API. This model can develop these services in pre-existing and new buildings. In this context, experts and professional in each subsystem are requested to participate in the design of new services.

To test the functionality of the model and to show how it can be implemented in real facilities, an experimental work was conducted in a residential home. In this context, several examples of fog and edge node implementations are described. Pattern recognition and decision trees methods were developed to show the potential of artificial intelligence in designing IoT solutions. The results of the installed services show that edge and fog node deployment produces the expected benefits: integration and interoperability.

The presented work demonstrates how edge and fog paradigms can be integrated into a new architecture where their benefits can be reinforced, thus expanding the field of application. The main scientific contributions of this architecture are the integration of paradigms, interoperability of the technology and the facilities it provides to develop AI services. All these improvements are shown in different examples with the developed experiments. Specific optimisation and improvement will be developed in future works. In addition, new control rules using machine-learning platforms and AI paradigms will ensure that new smart services can be created and improved.

Author Contributions: All the authors wrote the paper and read and approved the final manuscript.

Funding: This research was supported by the Industrial Computers and Computer Networks programme (I2RC) (2017/2018) funded by the University of Alicante, Wak9 Holding BV company under the eo-TICC project, the Valencian Innovation Agency under scientific innovation unit (UCIE Ars Innovatio) of the University of Alicante and by the Spanish Research Agency (AEI) and the European Regional Development Fund (ERDF) under the project CloudDriver4Industry TIN2017-89266-R.

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

References

1. Ren, J.; Pan, Y.; Goscinski, A.; Beyah, R.A. Edge computing for the Internet of things. *IEEE Netw.* **2018**, *32*, 6–7. [[CrossRef](#)]
2. Satyanarayanan, M. The Emergence of Edge Computing. *Computer* **2017**, *50*, 30–39. [[CrossRef](#)]
3. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [[CrossRef](#)]
4. Östberg, P.O.; Byrne, J.; Casari, P.; Eardley, P.; Fernandez Anta, A.; Forsman, J.; Kennedy, J.; Duc, T.L.; Mariño, M.N.; Loomba, R.; et al. Reliable capacity provisioning for distributed cloud/edge/fog computing applications. In Proceedings of the 2017 European Conference on Networks and Communications (EuCNC), Oulu, Finland, 12–15 June 2017; pp. 1–6.
5. Xu, J.; Palanisamy, B.; Ludwig, H.; Wang, Q. Zenith: Utility-Aware Resource Allocation for Edge Computing. In Proceedings of the IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, USA, 25–30 June 2017; pp. 47–54.
6. Wang, N.; Varghese, B.; Matthaiou M.; Nikolopoulos, D.S. ENORM: A Framework For Edge NODe Resource Management. *IEEE Trans. Serv. Comput.* **2017**. [[CrossRef](#)]

7. Liu, H.; Eldarrat, F.; Alqahtani, H.; Reznik, A.; de Foy X.; Zhang, Y. Mobile Edge Cloud System: Architectures, Challenges, and Approaches. *IEEE Syst. J.* **2017**, 1–14. [[CrossRef](#)]
8. Chen, X.; Jiao, L.; Li W.; Fu, X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE/ACM Trans. Netw.* **2016**, *24*, 2795–2808. [[CrossRef](#)]
9. Wang, L.; Jiao, L.; Kliazovich, D.; Bouvry, P. Reconciling task assignment and scheduling in mobile edge clouds. In Proceedings of the IEEE 24th International Conference on Network Protocols (ICNP), Singapore, 8–11 November 2016; pp. 1–6.
10. Ahn, G.; Park, Y.-J.; Hur, S. Probabilistic Graphical Framework for Estimating Collaboration Levels in Cloud Manufacturing. *Sustainability* **2017**, *9*, 277. [[CrossRef](#)]
11. Hoefler, C.N.; Karagiannis, G. Taxonomy of cloud computing services. *IEEE Globecom Workshops* **2010**, 1345–1350. [[CrossRef](#)]
12. Ghazouani, S.; Slimani, Y. A survey on cloud service description. *J. Netw. Comput. Appl.* **2017**, *91*, 61–74. [[CrossRef](#)]
13. Ruan, J.; Chan, F.T.S.; Zhu, F.; Wang, X.; Yang, J. A Visualization Review of Cloud Computing Algorithms in the Last Decade. *Sustainability* **2016**, *8*, 1008. [[CrossRef](#)]
14. Bokhari, M.U.; Makki, Q.; Tamandani, Y.K. A Survey on Cloud Computing. *Big Data Anal. Adv. Intell. Syst. Comput.* **2018**, *654*, 149–164.
15. Mukherjee, M.; Matam, R.; Shu, L.; Maglaras, L.; Ferrag, M.A.; Choudhury, N.; Kumar, V. Security and Privacy in Fog Computing: Challenges. *IEEE Access* **2017**, *5*, 19293–19304. [[CrossRef](#)]
16. Xiao, M.; Zhou, J.; Liu, X.; Jiang, M. A Hybrid Scheme for Fine-Grained Search and Access Authorization in Fog Computing Environment. *Sensors* **2017**, *17*, 1423. [[CrossRef](#)] [[PubMed](#)]
17. Zhang, Y.; Huang, H.; Xiang, Y.; Zhang, L.Y.; He, X. Harnessing the Hybrid Cloud for Secure Big Image Data Service. *IEEE Internet Things J.* **2017**, *5*, 1380–1388. [[CrossRef](#)]
18. Panda, S.K.; Jana, P.K. SLA-based task scheduling algorithms for heterogeneous multi-cloud environment. *J. Supercomput.* **2017**, *73*, 2730–2762. [[CrossRef](#)]
19. Faniyi, F.; Bahsoon, R. A Systematic Review of Service Level Management in the Cloud. *ACM Comput. Surv.* **2016**, *48*, 43. [[CrossRef](#)]
20. Singh, S.; Chana, I. QoS-Aware Autonomic Resource Management in Cloud Computing: A Systematic Review. *ACM Comput. Surv.* **2016**, *48*, 42. [[CrossRef](#)]
21. Xu, H.; Qiu, X.; Sheng, Y.; Luo, L.; Xiang, Y. A Qos-driven Approach to the Cloud Service Addressing Attributes of Security. *IEEE Access* **2018**, *6*, 34477–34487. [[CrossRef](#)]
22. Khan, A.M.; Freitag, F. On Edge Cloud Service Provision with Distributed Home Servers. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Hong Kong, 11–14 December 2017.
23. Mora, H.; Colom, J.F.; Gil, D.; Jimeno-Morenilla, A. Distributed computational model for shared processing on Cyber-Physical System environments. *Comput. Commun.* **2017**, *111*, 68–83 [[CrossRef](#)]
24. Mora, H.; Signes-Pont, M.T.; Gil, D.; Johnsson, M. Collaborative Working Architecture for IoT-Based Applications. *Sensors* **2018**, *18*, 1676. [[CrossRef](#)] [[PubMed](#)]
25. Al-Ayyoub, M.; Al-Quraan, M.; Jararweh, Y.; Benkhelifa, E.; Hariri, S. Resilient service provisioning in cloud based data centers. *Future Gener. Comput. Syst.* **2018**, *86*, 765–774. [[CrossRef](#)]
26. Cardoso, I.D.; Barraca, J.P.; Gonçalves, C.; Aguiar, R.L. Seamless integration of cloud and fog networks. *Int. J. Netw. Mgmt.* **2016**, *26*, 435–460. [[CrossRef](#)]
27. Farahzadi, A.; Shams, P.; Rezazadeh, J.; Farahbakhsh, R. Middleware technologies for cloud of things—A survey. *Digit. Commun. Netw.* **2018**, *4*, 176–188. [[CrossRef](#)]
28. Borgia, E. The Internet of Things vision: Key features, applications and open issues. *Comput. Commun.* **2014**, *54*, 1–31. [[CrossRef](#)]
29. Marco, C. Internet of things for Energy efficiency of buildings. *Int. Sci. J. Archit. Eng.* **2014**, *2*, 24–28.
30. i-SCOOP. Facility Management and Smart Buildings: Smart Data, Insights and Integration. Available online: <https://www.i-scoop.eu/Internet-of-things-guide/facility-management-iot-smart-buildings/> (accessed on 16 June 2018).
31. Intel Building Management Platform. Smart Building with Internet of Things Technologies. Available online: <https://www.intel.com/content/www/us/en/smart-buildings/overview.html> (accessed on 16 June 2018).

32. Siemens. Digitalize Your Building. Available online: <https://www.siemens.com/global/en/home/company/topic-areas/intelligent-infrastructure/buildings/digital-buildings.html> (accessed on 16 June 2018).
33. Min, W.; Seung Ho, H.; Musharraf, A. An IoT-based energy-management platform for industrial facilities. *Appl. Energy* **2016**, *164*, 607–619.
34. FacilitiesNet. Edge Devices Are Spurring IoT Transformation of Buildings. Available online: <https://www.facilitiesnet.com/buildingautomation/tip/Edge-Devices-Are-Spurring-IoT-Transformation-of-Buildings--38738> (accessed on 16 June 2018).
35. Selahattin, G.; Nikolay, Z.; Kurt, G.; Berna, O. IoT in Action: Design and Implementation of a Building Evacuation Service. *J. Comput. Netw. Commun.* **2017**, *9*, 8595404.
36. Alaa, A.; Zaidan, A.A.; Zaidan, B.B.; Talal, M.; Kiah, M.L.M. A review of smart home applications based on Internet of Things. *J. Netw. Comput. Appl.* **2017**, *97*, 48. [CrossRef]
37. Weinreich, A. The Future Of The Smart Home: Smart Homes and IoT: A Century In The Making, Forbes. 2017. Available online: <https://www.forbes.com/sites/andrewweinreich/2017/12/18/the-future-of-the-smart-home-smart-homes-iot-a-century-in-the-making/#5d01b9f857ac> (accessed on 10 October 2018).
38. Stojkoska, B.L.R.; Trivodaliev, K.V. A review of Internet of Things for smart home: Challenges and solutions. *J. Clean. Prod.* **2017**, *140*, 1454–1464. [CrossRef]
39. Hui, T.K.L.; Sherratt, R.S.; Díaz-Sánchez, D. Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies. *Future Gener. Comput. Syst.* **2017**, *76*, 358–369. [CrossRef]
40. Tolga, K.; Esra, B. An Investigation on Internet of Things Technology (IoT) In Smart Houses. *Int. J. Eng. Res. Dev.* **2017**, *9*, 196–207.
41. Pan, J.; McElhannon, J. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.* **2018**, *5*, 439–449. [CrossRef]
42. Ferrández-Pastor, F.J.; García-Chamizo, J.M.; Nieto-Hidalgo, M.; Mora-Martínez, J. Precision Agriculture Design Method Using a Distributed Computing Architecture on Internet of Things Context. *Sensors* **2018**, *18*, 1731. [CrossRef] [PubMed]
43. Chang, H.L.; Wang, C.G.; Wu, M.T.; Tsai, M.H.; Lin, C.Y. Gateway-Assisted Retransmission for Lightweight and Reliable IoT Communications. *Sensors* **2016**, *16*, 1560. [CrossRef] [PubMed]
44. Cruz-Piris, L.; Rivera, D.; Marsa-Maestre, I.; de la Hoz, E.; Velasco, J.R. Access Control Mechanism for IoT Environments Based on Modelling Communication Procedures as Resources. *Sensors* **2018**, *18*, 917. [CrossRef] [PubMed]
45. Atmoko, R.A.; Riantini, R.; Hasin, M.K. IoT real time data acquisition using MQTT protocol. *J. Phys. Conf. Ser.* **2017**, *853*, 012003. [CrossRef]
46. Babovic, Z.B.; Protic, J.; Milutinovic, V. Web Performance Evaluation for Internet of Things Applications. *IEEE Access* **2016**, *4*, 6974–6992. [CrossRef]
47. MQTT-org. MQTT Protocol. 2018. Available online: <https://www.mqtt.org> (accessed on 21 June 2018).
48. MQTT. MQTT Essentials Part 6: Quality of Service. 2018. Available online: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels> (accessed on 16 June 2018).
49. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
50. Miloud-Aouidate, A.; Baba-Hali, A.R. Survey of Nearest Neighbor Condensing Techniques. *Int. J. Adv. Comput. Sci. Appl.* **2011**, *2–11*, 59–64.
51. Amazon AWS core IoT. 2018. Available online: <https://aws.amazon.com/es/iot-core/features/> (accessed on 16 June 2018).
52. Microsoft. Connect, Monitor, and Manage Billions of IoT Assets. 2018. Available online: <https://azure.microsoft.com/en-us/services/iot-hub/> (accessed on 16 June 2018).
53. Ubidots. Data Drives Decision. 2018. Available online: <https://ubidots.com/> (accessed on 16 June 2018).
54. Thingspeak. Understand Your Things. 2018. Available online: <https://thingspeak.com/> (accessed on 16 June 2018).

