University of Salamanca

DOCTORAL THESIS

Adaptive model for monitoring and control of dynamic IoT networks

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Departament of Computer Science and Automation

Faculty of Science

Author:

Roberto Casado Vara

Supervisor:

Prof. Dr. Juan Manuel Corchado Rodríguez

Cosupervisor:

Dr. Javier Prieto Tejedor

Salamanca, 2019

Statement of Authorship

D. Roberto Casado Vara, presents the thesis project entitled "Adaptive model for monitoring and control of dynamic IoT networks." to apply for the Doctorate Degree in Computer Engineering form the University of Salamanca, and states that it has been carried out under the direction of Dr. Juan Manuel Corchado Rodríguez, University Professor of the Department of Informatics and Automation Control of the University of Salamanca, and Dr. Javier Prieto Tejedor.

Salamanca, May 13, 2019

Author:

Roberto Casado Vara

Supervisors:

Prof. Dr. Juan Manuel Corchado Rodríguez Dr. Javier Prieto Tejedor

Abstract

Smart cities have emerged out of our need to lead more sustainable lives. Thus, these cities incorporate technology and innovation into their infrastructures, reducing energy consumption and CO_2 emissions. Many European countries have established regulations for energy consumption in public and private buildings. These regulations are just some of the measures undertaken to achieve the targets set out in the Kyoto Protocol which aim to reduce CO_2 emissions and enhance energy savings. One of the main motivations behind the European H2020 project is to reduce energy consumption, as a result, researchers focus on the design of smart buildings and cities. The smart buildings concept has benefited from the technology boom that resulted in the Internet of Things paradigm and more precise sensors and actuators that contribute to efficient energy consumption monitoring and control in intelligent buildings.

The present doctoral thesis defines algorithms and a layer for IoT architectures for optimized monitoring and control processes in dynamic Internet of Things networks deployed in smart buildings. The algorithms and the architecture are integrable and are designed to reduce energy consumption in smart buildings regardless of their size, and thus significantly reduce the energy they consume.

Modules containing the designed algorithms have been integrated into the developed architecture. Those algorithms process the data collected by the Internet of Things sensors, increasing the quality of data, detecting inaccurate sensors and predicting their future efficiency. Because of those actions the monitoring and control of intelligent buildings is improved, and this has an immediate effect on the energy efficiency of intelligent buildings. In addition, the architecture integrates control techniques that increase the fault tolerance of sensors and blockchain-based techniques that provide security and privacy to the users of intelligent buildings.

As a result, a modular and self-adaptive architecture is obtained, capable of optimizing the energy efficiency of intelligent buildings by keeping the sensor networks robust against failures and the collected information immutable and private.

Resumen

Las ciudades inteligentes o Smart cities, son el resultado de la necesidad cada vez más imperiosa de orientar nuestra vida hacia la sostenibilidad. Así, estas ciudades se sirven de infraestructuras, innovación y tecnología para disminuir el consumo energético y reducir las emisiones de CO_2. Muchos países europeos están regulando el consumo energético en los edificios públicos y privados. Esta medida vio la luz como una de las medidas para conseguir los objetivos establecidos en el Protocolo de Kioto y con objeto de reducir las emisiones de CO_2 y potenciar el ahorro energético. Dentro de las ciudades inteligentes y debido a la motivación de reducir el consumo energético llevado a cabo por proyectos como el H2020 europeo, nace el concepto de edificios inteligentes o Smart buildings. Estos edificios han aprovechado la explosión tecnológica que ha tenido el Internet de las cosas con sensores y actuadores más precisos para poder ser más eficientes en la tarea de monitorizar y controlar el consumo energético en los edificios inteligentes.

El objetivo de esta Tesis Doctoral es investigar en métodos de optimización de los procesos de monitorización y control de redes dinámicas dentro del Internet de las Cosas de los edificios inteligentes, a través del diseño de una arquitectura escalable y algoritmos integrados en ella que sean capaces de reducir de manera drástica el consumo energético de los edificios independientemente de su tamaño.

Gracias a la implementación de esta arquitectura en un escenario real, los sensores del Internet de las cosas recogen datos de forma continua y estos datos son procesados por los distintos algoritmos desarrollados en esta Tesis doctoral, mejorando la calidad de los datos, detectando sensores poco precisos y haciendo predicciones de la eficiencia de estos sensores, para así conseguir una mejora de la monitorización y el control de estos edificios, lo cual, tiene como consecuencia inmediata la reducción del consumo energético de los edificios inteligentes. Además, la arquitectura integra técnicas de control para aumentar la tolerancia a fallos de los sensores y técnicas basadas en *blockchain* para proporcionar seguridad y privacidad a los usuarios de los edificios inteligentes.

Como resultado, esta Tesis doctoral presenta una arquitectura modular y auto-adaptativa, capaz de lograr la optimización del consumo energético de los edificios inteligentes manteniendo las redes de sensores robustas frente a fallos y a la información recogida inmutable y privada.

Agradecimientos

Quiero aprovechar esta oportunidad para dar las gracias a todas las personas que, con su apoyo, han hecho posible que haya podido alcanzar esta gran meta.

En primer lugar, me gustaría agradecer a mis directores, Juan Manuel y Javier, por darme la oportunidad de formarme como investigador bajo su tutela, por sus consejos, paciencia y ayuda. Por aportarme su diferente punto de vista sobre la investigación, lo que me ha permitido avanzar con paso firme y formarme como investigador.

Quiero agradecer a mis compañeros del grupo BISITE, todos ellos me han ayudado a sentirme miembro del grupo y, de todos ellos, he conseguido aprender algo nuevo durante estos años. En especial a todos los compañeros de cafe con los que, durante estos años, he compartido esos momentos de "social benefits".

Gracias a mis padres y a mí hermano, por apoyarme en todo momento sin ninguna duda. A mi primo Javier, por creer en mi y darme el empujón cuando mas lo necesitaba. A mi novia, por estar a mi lado soportando todas mis frustraciones durante el doctorado. Y, por último, a mis amigos, por estar ahí para ayudarme a desconectar.

Contents

Lists of Figures

Lists of Tables

Abbreviations

AI	**A**rtificial **I**ntelligence
IoT	**I**nternet **o**f **T**hings
QoS	**Q**uality **o**f **S**ervice
PoW	**P**rove **o**f **W**ork
SoA	**S**ervice **o**riented **A**rchitecture
GUID	**G**lobal **u**nique **ID**entifier
WSN	**W**ireless **S**ensor **N**etwork
NCS	**N**etwork **C**ontrol **S**ystem
SME	**S**ized **M**edium **E**nterprises
GT	**G**ame **T**heory
HDFS	**H**adoop **D**istributed **F**ile **S**ystem
REST	**RE**presentational **S**tate **T**ransfer
HVAC	**H**eating **V**entilation **A**ir **C**onditioning

Young man, in mathematics you don't understand things. You just get used to them. (John von Neumann)

Chapter 1

Introduction

VNiVERSiDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Chapter 1

Introduction

1.1 Introduction

This doctoral thesis comprises research in the field of efficient energy use, that has been conducted over a period of two years, within the BISITE research group of the University of Salamanca. The knowledge acquired during this time has made it possible to develop the hypothesis proposed in this doctoral thesis.

The hypothesis put forward in this thesis is that it is possible to further optimize the energy efficiency of a smart building regarding current monitoring and control systems. Here we propose an improved IoT architecture and several novel algorithms that allow to reduce consumption without having to make any significant changes in the infrastructure of the smart building. This thesis makes significant progress in the field of energy saving by improving a series of steps of monitoring and control algorithms, guaranteeing highly improved current state monitoring and control in the Internet of Things (IoT) network.

The rest of the Chapter is organized as follows: problem description and motivation are presented in section 1.2. Section 1.3 shows research hypothesis and objectives of this work, Methodology are presented in section 1.4 and finally, section 1.5 shows the structure of this dissertation.

1.2 Problem description and motivation

The scientific community is interested in the Smart Building concept because it sees it as a better alternative to the traditional building management system which Smart Buildings will soon replace. Regulatory authorities are also implementing laws that encourage the adoption of smart buildings. Building contractors have begun to assume

a key role in constructing technology-driven buildings, because technology increases the value of their properties and makes the buildings more energy efficient [1].

The Smart Building concept can be defined as a set of communication technologies that are designed to enable different objects, sensors and functions within a building to communicate and interact with each other and also to be managed, controlled and automated remotely [2]. Indeed, technologies help connect a variety of subsystems that originally operated independently. Automated processes allow to control the building's operations including HVAC (Heating, Ventilation, Air Conditioning), lighting, security and other systems [3]. The scope of the Smart Building concept extends to the majority of objects found in a household, ranging from windows and elevators to vehicle charging and its main focus is to achieve the energy efficiency of those objects. However, the two categories with greatest potential for energy optimization are:

- Smart lighting systems that adjust the level of light according to the time of day and other smart elements like windows. The values provided by occupancy sensors and space management are also taken into account when calculating the required level of lighting.
- Smart HVAC systems are associated with different types of sensors and the underlying technology makes them capable of adjusting their parameters quickly and automatically.

The Smart Home and Smart Building concepts are complementary in terms of the technologies they leverage and The same regulations apply to them in terms of the initial implementation of applications, facilities and services [4]. The benefits of Smart Buildings are efficiency-related:

- **Energy efficiency.** Technology-driven energy management helps reduce electricity consumption within households and building facilities. The idea is to control energy consumption by activating/deactivating the lighting and HVAC systems and any other energy-consuming appliances by automatically communicating with smart objects, like smart windows or presence detectors, in real time, and remotely if necessary.
- **Improved safety and security.** This has been the initial motivation behind the implementation of smart systems in a building. The growing trend in research

[1] IDATE DigiWorld, Smart home and smart building market, April 2017
[2] http://blog.buildout.com/smart-build-technologyoffices/
[3] https://www.savemoneycutcarbon.com/category/lighting-controls/
[4] http://www.buildings.com/articledetails/articleid/19537/title/how-smart-buildingssave-energy

on secure access control arises from the need to authenticate authorized persons or detect an intrusion. There is also a trend in the development of monitoring services that warn of emergency situations such as fires.

- **Employee productivity.** Smart solutions contribute to the users' comfort and well-being and provide greater convenience; they regulate the temperature in the workplace and monitor essential indicators such as air quality or humidity, which boost employee productivity.

Despite the advances in this field there is still room for improvement in the field of monitoring and control of smart buildings. It is necessary to design mechanisms that will optimize those processes. In this way, buildings will be greener, more economically attractive and more energy efficient. One of the approaches to optimizing the energy efficiency of smart buildings is improving the monitoring and control processes.

The solution proposed in this thesis optimizes a series of monitoring and control processes in smart buildings. The following questions motivated our research:

- Does the routing algorithm of IoT architectures address server energy costs?
- Are data collected by IoT devices reliable?
- Will IoT devices be accurate in the near future?
- Could control mechanisms be adopted in smart buildings with heterogeneous data?
- Could the robustness of the IoT network be improved to enhance its reliability?

In this thesis we intend to find answers to those questions by improving the existing IoT network monitoring and control algorithms in smart buildings and creating the necessary algorithms if that process does not exist in the state of the art.

1.3 Hypothesis and objectives

This research work develops a solution on the basis of the previously stated hypothesis, considering the research gaps in the existing monitoring and control techniques of the IoT network in smart buildings:

The initial hypothesis of the present research work is formalized in that it is possible to improve or optimize some of the current techniques and technologies for the monitoring and control of dynamic IoT networks in smart buildings.

Several gaps have been detected in some of the IoT network monitoring and control processes, that could be improved to achieve greater energy efficiency. We modify current IoT network monitoring and control processes and techniques with the aim of improving them. Moreover, we develop new techniques, technologies and algorithms. The distinctive feature of this work is the modular design of the developed energy optimization methods, this type of design makes them integrable with any IoT architecture, reducing the expenses involved in their implementation. Thus, the final outcome of this work is a new modular and feasible energy saving system designed for implementation in smart building's dynamic IoT networks.

The main objective of this thesis is to optimize energy use in smart buildings with dynamic IoT networks by developing modular techniques and algorithms as effective optimization mechanisms for temperature monitoring and control in dynamic IoT networks.

This research work comprehends several research areas such as monitoring and control of IoT networks and the theory and development of new algorithms based on mathematical concepts such as consensus, prediction of future states or complex networks. Prior to the development of our proposal it will be necessary to study existing IoT network monitoring and control techniques and technologies designed for smart buildings, identifying research gaps and areas that require improvement. We will approach the study of these techniques and technologies orienting the solution of the model specifically to solve the problems associated with energy consumption due to inefficient processes in the current techniques and technologies existing in the IoT networks in smart buildings. To successfully complete this research it is necessary to establish more specific objectives that are going to make it possible to achieve the main goal. These objectives are described below:

- To analyze specific problems associated with the monitoring and control of dynamic IoT networks in smart buildings to detect existing shortcomings related to energy savings.
- To perform a study of the techniques and technologies for the monitoring and control of IoT network in smart building with heterogeneous data.
- To identify gaps in the field of IoT network monitoring and control processes in smart buildings, propose improvements and techniques for optimized performance or new algorithms for the control of the IoT network. These algorithms have to be modular to integrate them as individual modules in any IoT architecture regardless of the characteristics or the topology of the smart building.

- To propose and design effective mechanisms as solutions to the gaps detected in IoT network monitoring and control in smart buildings. In this way proposing an improved IoT architecture layer that integrates a modular algorithm manager for use of energy consumption optimization algorithms in smart buildings.
- To implement the theory of some mathematical fields such as game theory, queuing theory, Markov chains or complex networks as the basis of the new algorithms developed in this work.
- To evaluate the new algorithms through simulations and implementation in real application environments

1.4 Methodology

It is necessary to define all the activities that are to be performed over the course of this research, from the research stage to the results validation stage, in order to ensure that the expected result are achieved in each stage. The chosen method is the one presented in [Reason and Bradbury, 2001], it is known as *action-research*. It is action-oriented and change-oriented, enabling it to focus on defined problems in order to produce knowledge from associated researches over an established period of time. This popular methodology allows for the practice of empirical research. It starts with the identification of a real problem, for which all possible hypotheses are enumerated. Then, these hypotheses are studied and one is selected on the basis of which a proposal is developed. Subsequently, the study focuses on verifying the selected hypothesis and a series of actions are taken that help to determine whether it is true or false. Finally, conclusions are drawn from the evaluation of the results of the research. To formalize this research model, a series of activities have been defined and briefly described below. These activities concord with the objectives of this research work:

1. Identification and description of the characteristics of the problem of optimization of energy consumption in IoT networks in smart buildings. During this activity the problem will be presented, the characteristics of the monitoring and control of the dynamic IoT networks in smart buildings are defined and different hypotheses are proposed for the total or partial solution of the problem.
2. Incremental study and review of the state of the art. Throughout this phase, the state of the art of the areas, technologies and developments related to the present research have been analyzed, which have allowed the obtaining of the theoretical framework and its possible developments. This phase has enabled the enrichment

of the knowledge of monitoring and control in IoT networks and has improved the scientific quality of the research work presented.

3. An iterative and progressive design of the proposed model of action. Starting with the information obtained in the previous activities a comprehensive model has been designed, it integrates all the necessary components that allow to develop innovative and original techniques and technologies for achieving greater energy efficiency in smart buildings' IoT network monitoring and control.

4. Development and integration of the model through new algorithms. In this phase the functionality, components, iterations, etc. of the new modular algorithms have been formalized and they are integrated in the developed architecture. The implementation of the proposed solution has made it possible to perform tests and obtain results that have served for the evaluation and formulation of conclusions.

5. Continuous dissemination of knowledge, results and experiences to the scientific community. This continuous activity throughout the research process has allowed for various publications in journals, and presentations at conferences and workshops that have validated and publicize the progress and partial results of the various milestones of research in expert areas with the consequent feedback to the work.

1.5 Structure of the thesis dissertation

This doctoral thesis is divided into seven chapters and several appendices. The structure of each of the chapters is described below.

Chapter 1 provides the introduction to this research. It describes the problems associated with the optimization of energy consumption in IoT network monitoring and control in smart buildings. The chapter points to the importance of designing new approaches to this problem, designated to make the current models more efficient. The objectives, hypothesis and motivation that have led to the development of this research work are presented. Finally, the applied research methodology is detailed and the memory structure is described briefly.

Chapter 2 reviews the state of the art. It begins by analysing the current state of the technology used in the monitoring and control of IoT networks in smart buildings. The chapter analyzes the different techniques and technologies that will serve as support for the development of new architectures and algorithms that will be designed to give a solution to the problem found.

Chapter 3 provides a description of the improved architecture with the 4 new blocks and the new routing algorithm. These four blocks make it possible to integrate the algorithms that have been developed in this research work. In addition, a new routing algorithm is presented to increase the performance of the architecture and, therefore, its energy efficiency.

Chapter 4 details the novel algorithms designed in this work. The four algorithms presented in this chapter aim to optimize or improve the gaps found in the literature on IoT network monitoring and control processes in smart buildings. In the chapter four algorithms are presented, the first algorithm improves data quality and identifies malfunctioning IoT devices. The second one predicts the future states of precision. The third one presents a new technique for transforming the heterogeneous data collected by an IoT network into homogeneous data, while the fourth algorithm uses the previous ones to design a distributed algorithm that improves the robustness of the network and thus is more energy efficient.

Chapter 5 describes the conducted case studies. To evaluate and prove the effectiveness of the new architecture and the proposed algorithms, a series of case studies have been designed.

Chapter 6 analyzes the results obtained in the five case studies described in chapter 5. The results presented in this chapter prove the effectiveness of algorithms designed to optimize the operation of IoT network monitoring and control systems.

Chapter 7 outlines the conclusions drawn from the work and describes ways in which it contributes to the state of the art. It also presents the future lines of research left open by this research work.

Appendix A contains the list of articles related to this work that have been presented during the PhD program research period in international journals, as well as book chapters. The research projects in which this student has been involved are also mentioned, and have, to some extent, contributed to the elaboration of parts of this PhD dissertation.

Finally, a list is presented with all the bibliographical references that have been used in this doctoral thesis and that have been referenced throughout this thesis dissertation.

Chapter 2

Technologies & methods for optimizing energy saving

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Chapter 2

Technologies & methods for optimizing energy saving

2.1 Introduction

This chapter presents the state of the art of the main technologies for the achievement of this work: internet of things, data quality, continuous-time accuracy states prediction, complex networks and blockchain.

The mathematical background required for the design and development of new algorithms is discussed. These algorithms will ease the monitoring and control tasks of IoT networks. In this way, new algorithms are proposed that are going to improve in many ways the efficiency of monitoring and control of current IoT networks.

The concept of Internet of things is defined below. This concept has attracted the researchers' interest. Therefore, there is extensive literature on the concept of IoT and its applications. However, in the definition of IoT, there is a common component for all authors, its orientation on services. IoT should provide a number of services to citizens and for this, in general, platforms have been developed to provide them for the different applications that are required.

One of the important issues in IoT networks is their monitoring and control. As a result, the surrounding environment of the IoT networks can be sensorized by providing a large amount of data. Therefore, by using this data it is possible to control and optimize the processes that are monitored by the IoT network. This chapter also presents a study on the IoT networks and their different monitoring and control techniques.

Blockchain technology is introduced in this chapter. This new technology presents a series of notable advantages such as the immutability of the data and the cryptographic security of the data. Many researchers are exploring the use of this new technology in conjunction with IoT. This chapter presents a study of the combination of IoT and

blockchain technologies, and how this combination increases the advantages offered by IoT technology.

Finally, the conclusions drawn from the review of these three main technologies are presented.

The rest of the Chapter is organized as follows: an Internet of things review on applications, architectures and emerging technologies like blockchain is presented in section 2.2. Section 2.3 shows the state of the art related with data quality in data collected by IoT networks, a summary about continuous-time Markov chains is presented in section 2.4, section 2.5 shows the information related with complex networks and graph theory. Finally, the current control systems for IoT networks is shown in section 2.6 and the conclusions are outlined in section 2.7.

2.2 Internet of things

The Internet of Things are entering the diary of many industry sectors. The IOT concept was coined by a member of the Radio Frequency Identification (RFID) development community in 1999, and it has recently become more relevant to the practical world largely because of the growth of mobile devices, embedded and ubiquitous communication, cloud computing and data analytics [Patel et al., 2016]. Internet of Things is refer to the general idea of things, especially everyday objects, that are readable, recognisable, locatable, addressable through information sensing device and/or controllable via the Internet, irrespective of the communication means (whether via RFID, wireless LAN, wide area networks, or other means). Everyday objects include not only the electronic devices we encounter or the products of higher technological development such as vehicles and equipment but things that we do not ordinarily think of as electronic at all, such as food, clothing, etc [Van Kranenburg and Bassi, 2012, Vermesan and Friess, 2013]. For example, the concept of “smart city” is emerging. Smart urban systems not only provide improvements in the quality of life of inhabitants, but also greatly increase asset efficiency by including smart transport systems (e.g. intelligent mobility, vehicle automation and traffic control), smart grids, public lighting management, traffic light management, waste management, environmental monitoring (e.g. in city vehicles to monitor environmental parameters), water management, surveillance/intelligence, intelligent services, and crowd detection (where the general public uses smart phones, portable and car-based sensors to collect and forward for aggregation of a variety of visual, signal and environmental data signals). Some of these services are known as ”smart street” services. In the short term, smart city industries cover five key areas: energy, water, mobility, buildings and government. The next

granular evolution of the intelligent city is the application of these concepts in a more limited physical environment. space, i.e. to the environments of commercial buildings. In fact, almost all applications of intelligent cities have a comparable value applicability to building management [Ray et al., 2016, Xu et al., 2016].

The IoT utilizes the Internet to incorporate heterogeneous devices with each other. In this regard and in order to facilitate the accessibility, all available devices should be connected to the internet. In order to achieve this target, sensors can be developed at different locations for collecting and analyzing data to improve the usage [Botta et al., 2016]. In this section we show a brief and concise review on IoT applications for smart cities.

2.2.1 IoT current applications for smart cities

The heterogeneity of different devices that can be connected to the Internet thanks to the IoT has led to the irruption of a vast range of application and architectures for smart cities related with identity, data authentication and integrity, privacy and secure communications. The main aims in the area of knowledge related to our work are summarized as the follows.

- **Smart cities and communities.** The application of IoT can lead to the creation of certain other services that interact with the surrounding environment. Therefore, it could introduces some chances for contextualization and geo-awareness. Moreover, collective intelligence will increase decision-making procedures and empower citizens [Alenezi et al., 2018]. In addition, a common middleware could be available for future smart cities services using IoT [Marques et al., 2019, Sikder et al., 2018, Tom et al., 2019]. It should be noted that sensor virtualization can be used to bridge the gap between current technologies and potential customers [Anthopoulos et al., 2019, Petrolo et al., 2014].

- **Smart homes and buildings.** Heterogeneous devices will allow the automation of common activities through the IoT platform in the home. In fact, by transforming objects into information devices that are connected to each other through the use of the Internet, services can be realized through web interfaces. A large number of applications in homes or intelligent buildings use sensor networks. These applications connect intelligent devices to the Internet to observe or control them remotely [Pilloni et al., 2018, Tatnall and Davey, 2019, Wu et al., 2018]. For example, intelligent lighting has been highly researched in recent years [Babar et al., 2019, Tao et al., 2018]. Nineteen percent of global electricity

consumption in cities if it is lighting that can cause six percent of greenhouse gas emissions [Fairchild, 2019, Moodie et al., 2018]. In this sense, up to forty-five percent of the energy needed for lighting could be saved by using intelligent lighting control mechanisms [Xie, 2019, Zou et al., 2018].

2.2.2 IoT architectures

Architectures are needed to represent, organize and structure the IoT in a way that enables it to function effectively. In particular, the distributed, heterogeneous nature of the IoT requires the application of hardware/network, software and process architectures capable of supporting these devices, their services, and the work flows they will affect. Here, several existing IoT architectures are explained [Kraijak and Tuwanut, 2015, Minoli et al., 2017].

- **Three-Level Architecture.** The three-level architecture is elementary for IoT, it has been implemented in a large number of systems. In [Chamoso et al., 2018, Cirani et al., 2014, Ray, 2018, Silva et al., 2018], different scalable architectures for IoT large-scaled network are presented. Usually, the following three levels can be identified in IoT networks: Internet-oriented, sensors and actuators, and knowledge. But in [Mahmoud et al., 2015], IoT architectures have the following three levels: Perception level, network level and application level.

- **SDN-Based Architecture.** Qin *et al.* [Muthanna et al., 2019, Qin et al., 2014, Sharma et al., 2018, Uddin et al., 2018], proposes an SDN-based IoT architecture to increase the quality of service (QoS). Casado-Vara *et al.* [Casado-Vara et al., 2018b], proposes an algorithm to increase the quality of data that can be used in IoT architectures, thus improving the QoS of IoT architectures.

- **QoS-Based Architecture.** Jin *et al.* [Abreu et al., 2017], propose four different IoT architectures that allow several applications for intelligent cities to include their QoS requirements. This architecture was improved in [Khan and Zeeshan, 2019, Matias et al., 2015, Rahimi et al., 2018, Sarwesh et al., 2019] to reduce the stress and congestion among nodes.

- **SoA-Based Architecture.** SoA is a component-based model that is designed to connect various services through applications and interfaces [Atzori et al., 2010, Gupta et al., 2018, Shashwat et al., 2018]. SoA architectures consists of four cooperating levels: 1-perception level, 2-network level, 3-service level and 4-application layer. In the fourth level of SoA-based architecture it is used to store and analyze data from IoT devices [Leu et al., 2014, Tiburski et al., 2015].

- **Cloud Things Architecture.** In Zhou *et al.* [Zhou et al., 2013] an IoT-enabled smart home scenario is proposed to analyze the IoT requirements. Hao *et al.* [Yue et al., 2014] proposes an architecture called Data Clouds, based on centralized information to increase the reliability of the new generation of Internet services. Nowadays, most of these architectures are implemented in the industry or smart cities [Manogaran et al., 2018, Samie et al., 2019].

Although these architectures are effective for the time being, we cannot be sure that for the challenges of the future they are reliable, and thus need to be re-examined. Integrating smart objects into physical infrastructure can improve flexibility, reliability and efficiency in infrastructures operation. These benefits can reduce cost and improve a lots of applications areas for the IoT such as smart infrastructure, healthcare, supply chains/logistics, social applications, surveillance, and so.

2.2.3 Data routing in IoT architectures: Queuing theory

Queuing theory is the mathematical study of queues or waiting lines within a system. This theory studies factors such as the average waiting time in queues or the working capacity of the system without collapsing. Within mathematics, queue theory is encompassed in operations research and is a very important complement to systems theory and control theory. It is thus a theory that finds application in a wide variety of situations such as business, commerce, industry, engineering, transport and logistics or telecommunications. They are formed due to a temporary imbalance between the demand for the service and the capacity of the system to supply it. The theory of tail formation is often too restrictive mathematically to be able to model all real situations worldwide. For example, mathematical models often assume the number of clients, or the capacity of the infinite queue, when it is evident that they must be limited. Alternative means of tail theory analysis usually consist of computer simulations or the analysis of experimental data.

In the context of computing and information and communication technologies, waiting situations within a network are more frequent. Thus, for example, the processes sent to a server for execution form waiting queues while they are not attended; the information requested, through the Internet, to a Web server can be received with delay due to congestion in the network; it is also possible to receive the line signal on which our busy mobile phone depends if the central office is collapsed at that moment, etc. As an example, telephonic networks are designed to handle the offered traffic intensity with only a small loss. The performance of the systems depends on whether the call is rejected, lost, etc. Normally overflow systems make use of alternative routes and even

these systems have a finite or maximum traffic carrying capacity. However, the use of queues allows systems to wait for customer requests until free resources are available. This means that if traffic intensity levels exceed the available capacity, customer calls would be lost. The discipline of queues determines the way in which customers' calls are handled. It defines how they will be served, the order in which they are served, and the way resources are divided among customers.

Queuing theory is a field of mathematics that has been studied extensively over the years. The queuing theory has different applications in mathematical models [Gnedenko and Kovalenko, 1989], computer applications [Kleinrock, 1976], linear statistical inference [Rao et al., 1973] and its applications and engineering systems [Blanchard et al., 1990]. Some of the newest work with queuing theory is in healthcare [Fomundam and Herrmann, 2007]. In this work authors present several applications of queuing theory to optimize the healthcare process. In another research, the relationship between telecommunications and queuing theory is shown. In their paper, the authors employ different queues and queue networks to increase the efficiency of telecommunication processes [Giambene, 2005]. Srivastava *et al.* present an in-depth study on the use of queuing theory together with big data technology for the optimization of the computational analysis of the data [Srivastava, 2018].

In recent years, some researchers have been optimizing the IoT network using queuing theory. Choi *et al.* present a $M|M|1$ queue system, in their approach they assume that the repository (server) is active when one (or more) vehicles are in motion [Choi et al., 2018]. Zhui *et al.* formulates the dynamic user scheduling and power allocation problem as a stochastic optimization problem with the objective to minimize the total power consumption of the whole network under the constraint of all users' long-term rate requirements [Zhai et al., 2018]. Chekired *et al.* proposes in their work a hierarchical fog servers' deployment at the network service layer across different tiers. Using probabilistic analysis models, they prove the efficiency of the proposed hierarchical fog computing compared with the flat architecture [Chekired et al., 2018].

2.2.4 Blockchain in the context of IoT

Industry and the research community have anticipated that blockchain technology is a disruptive technology that is ready to play an important role in the management, control and, most importantly, safety of IoT devices. This section describes how blockchain can be a key technology to provide viable security solutions to today's IoT security problems. This subsection first provides a brief overview of the block chain, then outlines the IoT security problems and challenges in open research for which blockchain can provide

solutions. The subsection also examines the literature of blockchain-based solutions for information security problems.

Blockchain is a distributed data structure that is replicated and shared among the members of a network [Bremer and Lehnhoff, 2017]. It was introduced with Bitcoin [Nakamoto, 2008] to solve the double-spending problem [Wiki, 2012]. As a result of how the nodes in Bitcoin (called miners) mutually validate the agreed transactions, Bitcoin's Blockchain establishes the owners and states what they own. A blockchain is built using cryptography. Each block is identified by its own cryptographic hash and each block refers to the hash of the previous block; this is how a link between the blocks is established, forming a blockchain [Antonopoulos, 2014] [Cardoso and Bordini, 2017]. For this reason, users can interact with blockchain by using a pair of public and private keys. In a blockchain, miners must agree on the transactions and the order in which they have occurred. Otherwise, the individual copies of this blockchain can diverge producing a fork; this means that miners have a different view of how the transactions have occurred, and it will not be possible to keep a single blockchain until the fork is not solved [Bui et al., 2018] [Capellari et al., 2018]. To achieve this, it is necessary to have a distributed consensus mechanism in every blockchain network [Casado-Vara and Corchado, 2018]. Blockchain's way of solving the fork problem is to link each blockchain node with the next block. This is done by finding a correct random number with SHA-256 [Monteriù et al., 2018] [Wiki, 2013] so that the number of zeros corresponds to the figure required by blockchain. Any node that solves this puzzle has generated the so-called proof-of-work (pow) and shapes the chain's next block. Since a one-way cryptographic hash function is involved, any other node can easily verify that the given answer satisfies the requirement [Pop et al., 2018]. Notice that a fork may still occur in the network when two competing nodes mine blocks almost simultaneously. Such forks are usually resolved automatically by the next block [Marín et al., 2015] [Becerra-Bonache and López, 2014]. The term "sidechain" was first described in the paper "Enabling Blockchain Innovations with Pegged Sidechains" [Medhi, 2002]. The paper describes "two-way pegged sidechains", a mechanism that allows the user to move the cryptocurrency within a sidechain, proving that some cryptocurrency that had previously been in a user's possession had been "locked".

2.2.5 Blockchain-based solutions for IoT

In the literature, research work on the safety of IoT and the blocking chain is limited, and most of the work focuses on harnessing blocking chain technology for the benefit of IoT in general. The authors in [Conoscenti et al., 2016] have categorized 18 cases of blocking chain use, of which four are for IoT . The four categories of IoT use

cases include immutable event logging and data access control management [Zyskind et al., 2015], trading of collected IoT data [Zhang and Wen, 2015], and management of symmetric and asymmetric keys for IoT devices [Axon, 2015]. The authors of [Friese et al., 2014] have outlined the challenges to identity in IoT. These challenges include primarily ownership and identity relationships, authentication and authorization, data governance, and privacy. In this subsection, we discuss how the block chain can be a key element in solving these challenges.

At the moment there are 5 billion IoT devices connected, and this number will continue to grow to 29 billion in 2022 [Panarello et al., 2018]. Each IoT device produces and exchanges data with the Internet. So, considering the large number of IoT devices, it's easy to understand that we're dealing with continuous massive production. In our opinion, blockchain represents the piece of the puzzle that solves the problems of privacy, large-amount of data+ and trust in IoT. As a blockchain it is decentralised, autonomous and without trustless features make it suitable to be applied in different scenarios such as smart cities [Casado-Vara et al., 2018a][Conoscenti et al., 2017], smart property [Herbert and Litchfield, 2015] and smart homes [Fromknecht et al., 2014][Caronni, 2000] as well. Useful applications of blockchain in IoT include, Miller *et al.* [Miller, 2018] proposes an interesting application of blockchain in IoT to solve the challenges of the supply chain and Novo *et al.* [Novo, 2018] shows an architecture for scalable access management in IoT.

Under IoT, the blockchain based on smart contracts is expected to play an important role in the management, control and security of IoT devices. In this subsection, we discuss and summarize some of the intrinsic characteristics of the blockchain that can be hugely useful for IoT in general, and for IoT safety in particular [Khan and Salah, 2018].

- **Address space.** Blockchain has a 160-bit address space,as opposed to IPv6 address space which has 128-bit address space [Antonopoulos, 2014]. A blockchain address is 20 bytes or a 160-bit hash of the public key generated by ECDSA (Elliptic Curve Digital Signature Algorithm). With 160-bit address, blockchain can generate and allocate addresses offline for around $1.46 \cdot 10^{48}$ IoT devices. The probability of address collision is approximately 10^{48},which is considered sufficiently secure to provide a GUID (Global Unique Identifier) which requires no registration or uniqueness verification when assigning and allocating an address to an IoT device.

- **Identity of things (IDoT).** Identity and Access Management(IAM) for IoT must address a number of challenging issues in an efficiently, secure, and trustworthy

manner. Blockchain has the ability to solve these challenges easily, securely, and efficiently. Blockchain has been used widely for providing trustworthy and authorized identity registration, ownership tracking and monitoring of products, goods, and assets. Blockchain also provides a trust worthy decentralized management, governance, and tracking at every point in the supply chain and lifecycle of an IoT device.

- **Data Authentication and Integrity.** By design, data transmit-ted by IoT devices connected to the blockchain network will always be cryptographically proofed and signed by the true sender that holds a unique public key and GUID, and thereby ensuring authentication and integrity of transmitted data. In addition, all transactions made to or by an IoT device are recorded on the blockchain distributed ledger and can be tracked securely.

- **Authentication, Authorization, and Privacy.** Blockchain smart contracts have the ability to provide a decentralized authentication rules and logic to be able to provide single and multi-party authentication to an IoT Device. Also, smart contracts can provide a more effective authorization access rules to connected IoT devices with way less complexity when compared with traditional authorization protocols like Role Based Access Management (RBAC), OAuth 2.0, OpenID, OMA DM and LWM2M. These protocols are widely used these days for IoT device authentication, authorization, and management. Moreover, data privacy can be also ensured by using smart contracts which set the access rules, conditions, and time to allow certain individual or group of users or machines to own, control, or have access to data at rest or in transit. The smart contracts can spell out also who has the right to update, upgrade, patch the IoT software or hardware, reset the IoT device, provision of new key pairs, initiate a service or repair request, change ownership, and provision or re-provision of the device.

- **Secure Communications.** IoT application communication protocols as those of HTTP, MQTT, CoAP, or XMPP, or even protocols related to routing as those of RPL and 6LoWPAN, are not secure by design. With blockchain, key management and distribution are totally eliminated,as each IoT device would have his own unique GUID and asymmetric key pair once installed and connected to the blockchain network. This will lead also to significant simplification of other security protocols as that of DTLS, with no need to handle and exchange PKI certificates at the handshake phase in case of DTLS or TLS (or IKE in case of IPSec) to negotiate the cipher suite parameters for encryption and hashing and to establish the master and session keys

2.3 Data Quality in IoT: Game theory-based consensus

IoT have become important in the last years and nowadays are present in practically all the sectors of our society [Haibo and Fang, 2015]. Their great capacity to gather data may facilitate the construction of smart environments, allowing for a flexible analysis of processes that occur in the environment and the services offered to users. There are many advances in IoT architecture however, the efficient management of the data generated by them is still a challenging aspect. Data management is complicated because the data may sometimes be inconsistent due to different reasons (i.e. it is difficult to determine if data is reliable or if sensors are accurate, etc.). Therefore, there is a growing need for new IoT architectures which would merge data from heterogeneous sensors, and intelligent management of the generated information.

The state of the art contains some architectures that allow to merge data from IoT [Gungor et al., 2009, Patel and Pandey, 2010]. There are also some novel frameworks that define methods for integrating dynamic and self-adaptable heterogeneous IoT [Tapia et al., 2010b], and manage data obtained from IoT [Rodríguez et al., 2015]. Other architecture proposals have demonstrated that the accuracy of IoT can be improved with the use of artificial neural networks [de Paz et al., 2013]. Another work presents a multi-agent system that automatically processes and merges information in heterogeneous distributed IoT [Bajo et al., 2015]. However, some frameworks are designed for very specific purposes and their functionality is limited [Alonso et al., 2013, Tapia et al., 2010a, 2009]. It is also difficult to merge and manage the data obtained from heterogeneous IoT [Bowman and Steinberg, 2008].

Game Theory (GT) is a branch of applied mathematics that is used to study how players can interact with each other with the aim of obtaining a fair and stable distribution of useful resources within the studied problem. This way, GT analyses the interaction between independent and self-interested players to find a solution to a particular problem. In [Saghezchi et al., 2017], the authors provide a general review of GT and demonstrates how it is applied to networking and signal processing problems.

On the other hand, a IoT is a network of sensors whose communications are transmitted by wireless signals. IoT are used to collect and study a wide range of magnitudes, such as sound, humidity, temperature and many others. IoT is the preferred as the architecture of the sensor systems, but they are susceptible to the disadvantages caused by the limited lifetime of the operation. Unlike other sensor networks with physical support (wire), the use of IoT is limited by amount of energy it can store, its calculation capacities, memory, information flow and communication distance (see [Zhou, J.; Mu, 2006]). Since in most cases sensors are deployed on a surface, it is difficult and expensive

to replace a faulty sensor manually by looking at the entire IoT. In addition, the sensors may lack global information of the entire network and the network topology can change with time. [Tapia et al., 2009] presents distributed sensor architectures, which allow the sensors to operate autonomously without the need for a central node, while [Corchado et al., 2010] provides a practical case of using heterogeneous distributed sensor networks.

On the other hand, IoT are a hot topic and there is a lot of research focused on IoT and GT, [Han, Z., Niyato, D., Saad, W., Başar, T., & Hjørungnes, 2011] is focused on finding innovative solutions to the challenges related to the next-generation IoT. Since GT is an ideal tool for designing efficient and robust distributed algorithms. The use of GT for the design and analysis of the IoT information has raised attention [Moura and Hutchison, 2017]. This survey looks at how GT is currently being used in IoT; from classic to evolved games, from cooperative to non-cooperative, and the NE in these cases.
In [Shi et al., 2012] a general classification of the different uses of GT in IoT is proposed; they are classified into the following groups: network management, communication, network security and applications. Our proposal can be included in the Applications category, within the Data Collection subgroup. This is one of the most important parts of a IoT, although in our case instead of focusing on the operation of the network itself and its information transmission capacities, we will focus on the collected data.

IoT research has many potential applications including environment and building monitoring, industrial process control, infrastructure and IoT security, and automatic transport (see [Kottapalli et al., 2003]). In this regard, [Lorincz et al., 2004] presents a study about the monitoring of public buildings and, the work proposes many research lines on the use of IoT for event detection. The authors of [Su et al., 2012] used this study as a basis for creating a fuzzy system for the temperature monitoring in a given area and for computing that area in the IoT. In this study they present a fuzzy algorithm for temperature control in which the distance between the sensors is a key issue.

In [Mishra et al., 2017], a method addressing the problem of energy efficiency using IoT, GT and ant colon based is presented. Initially, several clusters were formed without using the IoT, later on additional clusters were created using the coalitions formed by the game. The algorithm proposed in this game overcomes the spatial correlation of the harvested data generated by neighboring nodes to form coalitions within clusters. [Afsar, 2015] presents an approach to the problem of grouping sensor nodes using game coalitions and clustering theories (Coalition-games Theoretic Clustering). In [Schmidt, 2002], GT-based coalitions are presented for the analysis of the clustering methods-based economy. Finally, [Xu et al., 2013] presents a work that can be used in different

fields (economic, biological, wireless communications, etc.) where decisions are based on cooperative strategies.

The methods in the described literature, face different problems such as energy efficiency and economic analysis using IoT and GT respectively by forming coalitions with clustering techniques. Distributed sensor networks and depending on the network topology and sensor neighbourhood are also presented. In our work, coalitions of neighbours are created by using clustering techniques. This distributed and self-organized (overall temperature equilibrium arises from local game interactions between sensors of an initially disordered temperatures system) game is designed to provide reliability and robustness to the data collected by a IoT. It identifies sensors gathering defective or inaccurate measurements and detects areas with similar temperatures. This article tackles the problem of IoT data reliability from the point of view of game theory and probability, which is a novelty approach in this field.

2.4 Future IoT devices accuracy states prediction: Continuous-time Markov chains

With the development of communications techniques, network topologies and control methods, networked control systems (NCS) have received increasing attention in the past decades due to its widespread applications [Hespanha et al., 2007]. Meanwhile, because an IoT network is usually shared by multiple sensor, controller and actuator nodes, these IoT nodes collect data from a wide variety of buildings. There is a need for IoT network monitor and control to improve the detection of sensors that are collecting false data or malfunctioning [Mo et al., 2010]. This paper presents a new predictive feedback control algorithm for handling the predictive management of a huge amount of IoT nodes. There is a need to implement a control system that monitors and controls the accuracy states of the IoT nodes. In this way it is possible to ensure confidence in the data collected by the IoT network. Discrete-time control mainly studies the performance of the system in a discrete-time interval rather than a continuous time interval. The discrete-time control problems for linear systems have been investigated such as linear systems [Amato et al., 2006] [Amato et al., 2010] [Polyakov et al., 2016]. Meanwhile, the studies on the discrete-time control of nonlinear system have also been carried out for triangular systems [Korobov et al., 2013], nonlinear dynamical networks [Hui et al., 2008], etc. Discrete-time control techniques have been applied for many practical applications, for instance, multiagents systems [Khoo et al., 2014] and secure communications [Perruquetti et al., 2008]. Feedback nonlinear systems representing a class of nonlinear control systems have been widely concerned [Li et al.,

2015] [Li and Yang, 2018]. The problem we want to deal with in this paper is the predictive maintenance of IoT networks in continuous-time. In this way it is possible to improve the reliability in the monitoring and control of IoT networks as it is done in continuous-time. By using continuous-time Markov chains to predict the future states of sensor accuracy, IoT networks can improve the data quality since they will always be working in the best possible condition.

2.5 IoT network slicing: Complex network and clustering

IoT networks usually have heterogeneous data, but the most of algorithms do their best with homogeneous data. Therefore, there is a need to be able to make the algorithms we apply for monitoring and control of IoT networks can have as input homogeneous data. To do so, there is a need for theories, models, mechanisms, methodologies and tools that can develop a system capable of reorganizing and adapting itself to possible future changes in the environment. For this reason, the present work proposes a combination of technologies, such as graph and complex network theory, clustering techniques and game theory with IoT, such as the key technological context to confront the existing needs within smart building environments.

This section summarizes the state of the art related to the use of theoretical mathematical techniques such as complex networks and graph theory in the solution of real problems. It also shows the state of the art of clustering techniques used for supervised learning methods to find the solution to data classification problems.

2.5.1 Complex networks and graph theroy

A graph is a mathematical representation of a network and it describes the relationship between vertices and edges. Graph theory is used to represent real-life phenomena, but sometimes graphs are not able to properly represent many phenomena because uncertainty of different attributes of the systems exists naturally. Many real-world phenomena provided motivation to define the fuzzy graphs. Kauffman [Kaufmann, 1973] introduced fuzzy graphs using Zadeh's fuzzy relation [Zadeh, 1971]. Fuzzy-graph theory is growing rapidly, with numerous applications in many domains, including networking, communication, data mining, clustering, image capturing, image segmentation, planning, and scheduling. Graphs are widely used to model various structured data, including road maps [Beasley and Christofides, 1997], social networks [Watts et al., 2002], and molecular structures [Mahé et al., 2005, Marin et al., 2008]. Due to the extensive applications of graph data, considerable efforts have been made to develop techniques

for effective graph data management and analysis, such as graph mining [Xuan et al., 2015, Yan and Han, 2002], graph matching [Conte et al., 2004], and graph similarity search [Chen et al., 2016, Zhao et al., 2012]. Sampathkumar [Sampathkumar, 2006] introduced the notion of graph structures. Graph structures are the generalization of graphs and widely useful in the study of some structures, like graphs, signed graphs, semigraphs, edge-colored graphs, and edge-labeled graphs. Graph structures are very useful in the study of different domains of computer science and computational intelligence.

2.5.2 Clustering techniques

In recent years, the automation of data collection and recording has involved an avalanche of information on different types of systems [Bollen et al., 2009, Golder and Macy, 2011]. As a consequence, many methodologies aimed at organizing and modeling data have been developed. Such methodologies are motivated by their widespread application in diagnosis, education, forecasting, and many other domains. The definition, evaluation and application of these methodologies are all part of the machine learning field, which became a major subarea of computer science and statistics due to their crucial role in the modern world. Machine learning encompasses different topics such as regression analysis [Wang et al., 2010], feature selection methods [Blum and Langley, 1997], and classification [Witten et al., 2016]. The latter involves assigning classes to the objects in a dataset. Three main approaches can be considered for classification: supervised, semi-supervised and unsupervised classification. Clustering methods are generally more demanding than supervised approaches, but provide more insights about complex data. This type of classifiers constitute the main object of the current work.

In [Kinnunen et al., 2011], a comparative analysis of clustering methods was performed in the context of text independent speaker verification task, using three dataset of documents. Two approaches were considered: clustering algorithms focused in minimizing a distance based objective function and a Gaussian models-based approach. The following algorithms were compared: k-means, random swap, expectation-maximization, hierarchical clustering, self-organized maps (SOM) and fuzzy c-means. In [de Souto et al., 2008], five clustering methods were studied: k-means, multivariate Gaussian mixture, hierarchical clustering, spectral and nearest neighbor methods. Four proximity measures were used in the experiments: Pearson and Spearman correlation coefficient, cosine similarity and the euclidean distance. The algorithms were evaluated in the context of 35 gene expression data from either

Affymetrix or cDNA chip platforms, using the adjusted rand index for performance evaluation. In [Costa et al., 2004], experiments were performed to compare five different types of clustering algorithms: CLICK, self organized mapping-based method (SOM), k-means, hierarchical and dynamical clustering. Data sets of gene expression time series of the Saccharomyces cerevisiae yeast were used. Many different types of clustering methods have been proposed in the literature [Agrawal et al., 1998, Guha et al., 1998]. Despite such a diversity, some methods are more frequently used [Wu et al., 2008]. Several taxonomies have been proposed to organize the many different types of clustering algorithms into families. While some taxonomies categorize the algorithms based on their objective functions [Jain et al., 2004], others aim at the specific structures desired for the obtained clusters (e.g. hierarchical) [Fraley and Raftery, 1998].

In recent years, the efficient handling of high dimensional data has become of paramount importance and, for this reason, this feature has been desired when choosing the most appropriate method for obtaining accurate partitions. So far, we have discussed the application of clustering algorithms on static data. Nevertheless, when analyzing data, it is important to take into account whether the data are dynamic or static. Dynamic data, unlike static data, undergo changes over time. Some kinds of data, like the network packets received by a router, IoT temperature stream and credit card transaction streams, are transient in nature and they are known as data stream.

2.6 IoT monitoring and control: Improving robustness of the network

A NCS is a control system wherein the control loops are closed through a communication network [Zhang et al., 2016]. Their advantages include low installation and maintenance costs, flexibility and reduced wiring [Ge et al., 2017]. These benefits make NCSs applicable to a wide range of fields [Qiu et al., 2016]. However, NCS's weaknesses are caused by delays in random communications and packet loss. In terms of current industrial applicability, traditional point-to-point centralized control is unsuitable since it does not meet the new requirements such as modularity, decentralized/distributed control, quick and easy maintenance and low cost. For these reasons, in recent years NCS has been a major focus of attention in both academic research and industrial applications, contributing to significant progress in this field [Hespanha et al., 2007]. IoT networks usually consist of many sensors, as well as controller and actuator nodes. These distributed IoT nodes compete to send their data to the network. There is a need to implement a control system that monitors and controls the sending of packets from the IoT nodes to the network. In addition, IoT networks generate large amounts

of data continuously. The volume of data makes it very difficult to monitor and control IoT networks. To control the IoT network it is necessary to search within the databases. Causing a delay in the functioning of the control system within the IoT network. This also entails high levels of consumption of energy and resources.

Although the problem of the limitation of communications in networks has been well studied, there are relatively few works on the optimization of queuing analysis. Furthermore, none of those works incorporate properties of the network into the control system [Sun et al., 2018] [Liu et al., 2018] [Casado-Vara et al., 2018c]. The problem remains unresolved; the properties of IoT networks continue to affect system performance to a large degree. Thus, these properties must be taken into consideration by the control algorithm used by the networks [Tan et al., 2015]. Most of the research is based on optimizing energy consumption for improved performance of NCSs. However, the use of NCSs with IoT network properties, including queue analysis, service rate and packet dropout, must be subjected to further research. In order to make the monitoring and control of the IoT network more effective, some researchers proposed new techniques that increase the speed at which big data databases are searched. Zhou *et al.* designed a framework to bridge multi-target query needs between users and the data platform, including required query accuracy, timeliness, and query privacy constraints [Zhou et al., 2018]. Another research proposed the use of binary hashing for greater search speed; Cao *et al.* reviewed and compared those hash techniques through different experiments [Cao et al., 2018].

2.7 Conclusions

This chapter has reviewed the state of the art of the IoT architectures and their monitoring and control. Based on the state of the art presented in this chapter, we have found that existing IoT solutions have a gap in the optimization of their operations focused on energy savings. This Doctoral Thesis aims to solve these problems by further investigating on the following areas:

- The IoT applications and architectures those are most frequently used today have been revised. One of the gaps in the literature we have found is that routing systems are based on the characteristics of the network itself. But we have proposed a new routing system based on network characteristics and optimized for blockchain integration. Blockchain has allowed me to solve some security and identity problems that we had found in the state of the art study related to blockchain architectures.

- Consensus techniques based on game theory make it possible to reach agreements between different players according to the interests of the majority. These agreements are benefits for all the different players which allows to reach the best possible solution. Applied to IoT networks, the consensus based on game theory will improve the quality of the data collected by IoT networks by making coalitions between IoT devices to find the wrong data and correct it.

- Continuous-time Markov chains will improve the maintenance of the IoT network as it will be possible to predict in continuous time the accuracy state of the IoT devices, being able to react promptly to malfunction of the IoT network devices.

- The use of complex network and clustering techniques in the study of data collected by IoT network will allow to find groups of IoT devices that collect the same type of data according to the characteristics of the IoT devices and the topology of the IoT network.

- Control algorithms allow the simulation of different conditions, in several contexts and under uncontrolled conditions. If the simulations show an adequate results, a control algorithm can be designed that control an IoT network. This allows to move the actions taken in the simulation (e.g., in simulink) to a real context, providing pretty similar results.

Chapter 3

Adaptive management blocks middleware layer

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Chapter 3

Adaptive management blocks middleware layer

3.1 Introduction

This present chapter describes the overall architecture proposed to integrate algorithms which optimize the monitoring and control task in smart buildings. It can be implemented in any smart building regardless of the building's technical characteristics. The objective is to make the proposed architecture adaptable to any environment and to spare the user from having to make any configurations in the system. To achieve this, the architecture has to be self-adaptive and also dynamic since the data collected in real-time, on the factors that influence consumption varies continuously, while the number of users is also variable over time. this is the only way to make satisfactory decisions supported by control algorithms in terms of energy, providing a non-invasive mechanism for energy saving. The main contributions of this chapter are:

- The architecture presented in this chapter is the one used as a basis that defines the model and the technologies to be incorporated into the system that will later be applied in each energy optimization case study or simulation. Notice that the architecture used in this research work is a typical IoT architecture in which the data analysis layer is added with the 4 blocks for the intelligent management of the proposed new algorithms. The main aim of this research work is not the development of an architecture, but the design and development of algorithms more efficient than the current ones to improve the monitoring and control of smart buildings.
- This chapter shows the difference between the concept of an architecture and the concept of a system. The architecture defines the model to be followed (i.e., the structuring of the system outlined in the initial stages of research, which sets the

framework for the next stages of development). On the other hand, a system is the result of implementing the architecture in a context. Therefore, as many systems as necessary can be developed following the structure of the designed architectures. Thus, we can have a system for the management of a building or a platform for each fault in the building.

- The proposed architecture is based on three essential parts. A set of hardware technologies, devices and actuators, which make it possible to collect context data, a smart controller which ingest the data into the architecture. A set of own-designed algorithms which are encapsulated in a modular set of blocks that make up the systems. Finally, a service layer which enables the interaction with the people.

he rest of the chapter is orgnaized as follows: the first part of the chapter focuses on that set of hardware technologies, and the data ingestion that should be Incorporated for the system to correctly fulfill its purpose is detailed in section 3.2.1. The second part of this chapter focuses on how these algorithms detailed in 4 are integrated within the blocks in order to perform energy optimization in smart buildings, the interaction with the blockchain and the routing protocol optimized for this architecture are presented in section 3.2.2.

3.2 IoT architecture overview

This subsection describes the novelties of this work and the final architectures which integrates them. Our main focus was the design of a new self-adaptive control algorithm for the monitoring and control of block flow from the IoT devices to the blockchain (see Fig. 3.1). Also, we propose a new way of storing data in a big data ecosystem. The sensor ID, timestamp and query are stored in a hashmap in the blockchain. In this way, it is easier to search data what also contributes to optimized monitoring and control of the IoT devices. This architecture works as follows: 1) Smart devices collect data from the environment and start processing them. In the first layer, smart devices incorporate the edge computing paradigm which allows to execute smart contracts to insert the data from IoT devices into a sidechain (i.e., a permissioned blockchain designated for the sole use of smart devices). Once the smart devices have finished validating and inserting data into blocks, they execute a new smart contract that sends the blocks to the miners' network for inclusion in the blockchain. 2) Blockchain nodes (i.e., miners) receive blocks already built from the sidechain. Then, the blockchain miners' network validates the already built block in the sidechain and includes it in the blockchain. Thus,

the miners only have to calculate the hash that allows to add the node to the blockchain and computing power is not required to build the block. 3) Once data of the IoT devices is already in the blockchain, through a real time streaming this data is inserted into the big data layer. Once data is in HDFS (Hadoop Distributed File System) a smart contract is executed to send the addresses of the data stored in HDFS (since HDFS replicates the content and distributes it within the database) and the query to have access to the data (this query is built in MapReduce) that are stored in a hashmap. In this way, data are secured in the blockchain. Since they are stored in HDFS, their availability is high and the monitoring and control of the IoT devices is optimized by big data technologies (Business intelligence, data discovery, machine learning, etc.). The following paragraphs give an in-depth description of our proposal; the improved architecture and its functioning.

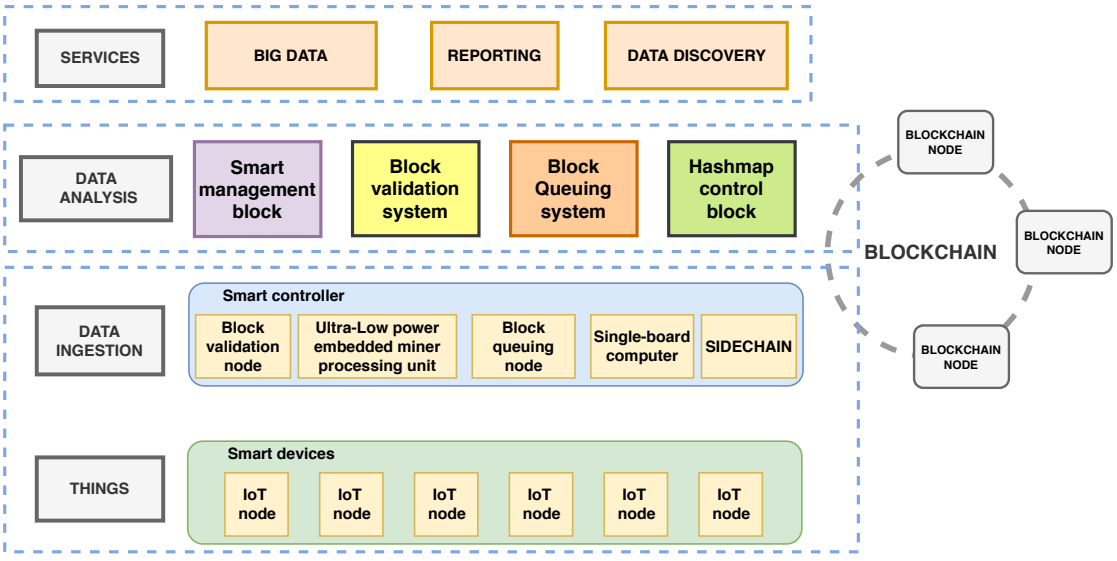

FIG. 3.1: **Main architecture.** This architecture improved the monitoring and control system of the blocks sent to the blockchain from the IoT nodes. Also, this architecture allows for faster search of data via hashmap in the blockchain.

3.2.1 Things and data ingestion layer

Fig. 3.2 shows how smart devices (i.e., Event producer layer) collect and pre-process (i.e., Pre-processing layer) data before sending them to the blockchain. Smart devices have the following layers: 1) Event producer layer. In this layer, IoT nodes collect data (temperature, humidity, etc.) and send them to the smart controllers for real time pre-processing. 2) Pre-processing layer. In this layer the single-board computer (in our case Raspberry Pi, although other similar devices could also be used) begins to build the block with the data collected by the IoT devices, once the block is built, the Raspberry Pi runs a smart contract that introduces data in the sidechain. To improve the processing

power of the Single-board computer, a device called the "Ultra-Low power embedded miner processing unit" is added via USB. This supports the Single-board computer in the construction of the blocks. This device improve the computational power of the smart device. This way, the smart controller can run complex algorithms in the single-board computer without the need for a computer with high computing power. The smart controllers are in the edge computing layer. Since the smart controllers process the data, they push the computing power away from the blockchain. Once the block is built, the block validation node asks the miners network for a validation request. Then, the block validation node (this node has the right to read the blockchain) tries to validate the block it has just built. In case the block is validated, the block validation node sends the block to the smart broker to manage the block submission queue to the miners' network, which inserts the block into the blockchain. In this way, the blockchain's miners' network only has to insert the blocks into the blockchain. Moreover, the smart controllers can run the mining algorithm to help with mathematical computations of the proof of work (PoW) of the next block that is attached to the blockchain.

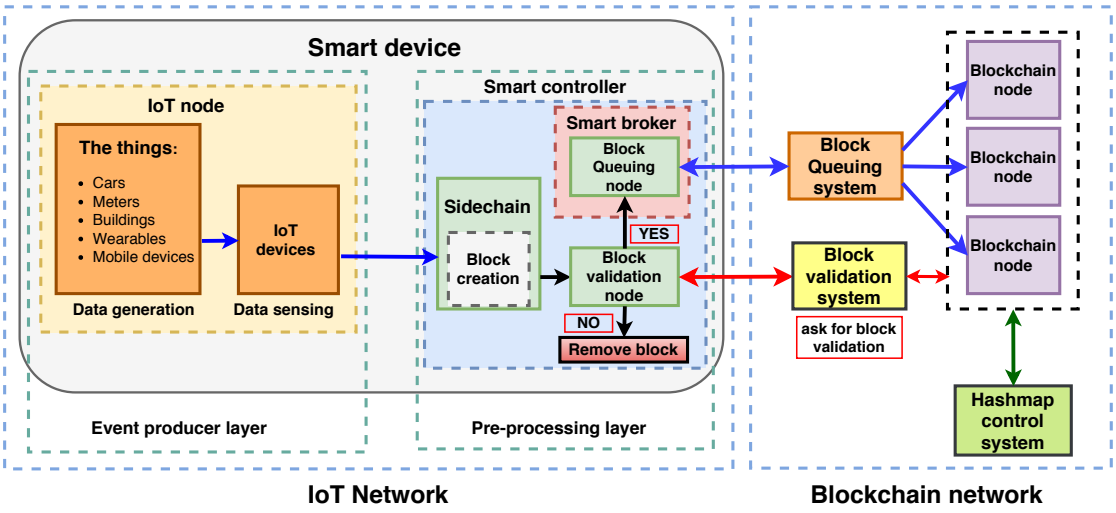

FIG. 3.2: **Edge computing overview.** IoT nodes collect data from the things, in this architecture this occurs in the event producer layer. Moreover, in the pre-processing layer the smart controllers build the blocks. With a smart contract this block is inserted into a permissioned blockchain. Once the block is in the blockchain, a smart broker who controls a block queuing system sends the sidechain blocks to the miners.

The following sections present a detailed description of the smart controller elements and their features, including: the sidechain, the block validation node, and the smart broker.

3.2.1.1 Sidechain

Sidechains are blockchains that are created according to the needs of the system. In this paper, the sidechains are created to store the data collected by the IoT devices. Fig. 3.3 shows the relationship between the blockchain and the sidechains. In the figure it can be observed that the IoT nodes send their information to the sidechains. The sidechains then share information (in the form of blocks) with the main blockchain. In this way, the sidechains can process the data of the IoT nodes in parallel, in the edge computing layer. These sidechains work in the same way as the main chain (blockchain). The IoT devices send the data to the smart controllers. Once the data is in the smart controller it reaches the sidechain. Then, the sidechain stores data until it has enough data to build a block. Blocks do not have a fixed size, but they are less than 1 MB. Once the block is created in the sidechain, data flow continues in the smart controller and the block validation node starts working.

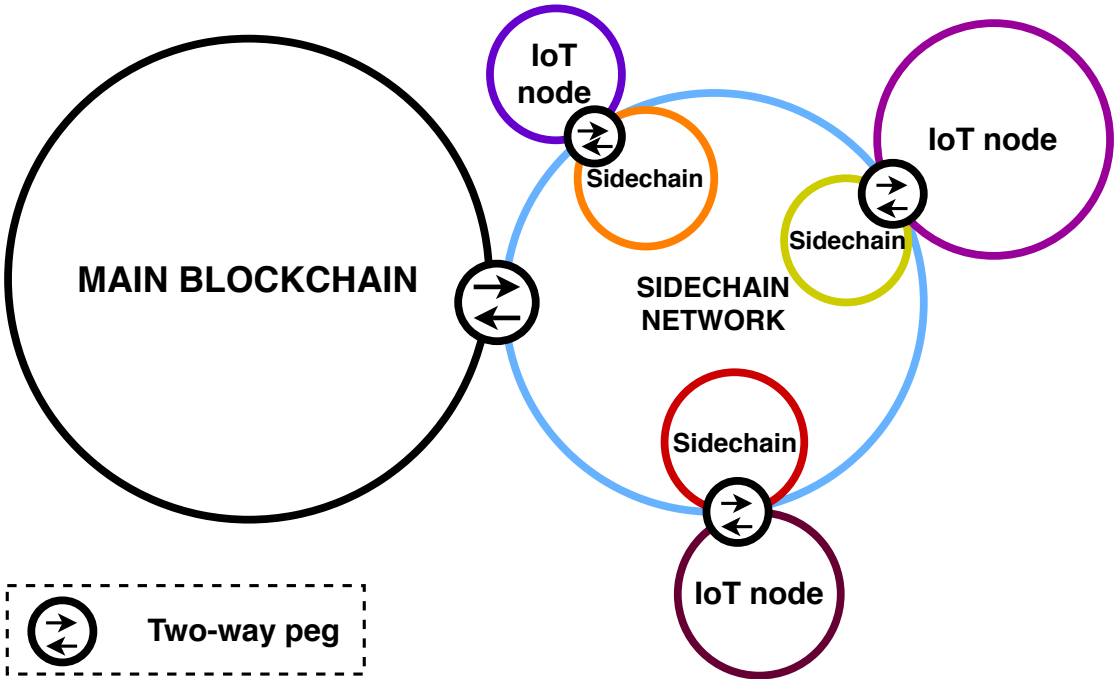

FIG. 3.3: **Blockchain and sidechain relationship.** Sidechains allow complex data updates in blockchain to be validated in a realistic environment (where current blockchain is at risk and utilizing actual network mining resources). If these updates fail, they can be removed; if they succeed, they can be incorporated into the mainchain.

3.2.1.2 Block validation node

Once the block is built in the sidechain, the smart controller has to validate the block before sending it to the miners' network. To do this, it uses the Block validation system

which has a node in each of the smart controllers that are connected to the central system in the miners' network. The block validation node is asked to validate the new block that has just been built in the sidechain. Then, it sends a copy of the block to the block validation system. This system validates that data in the block are correct and are within the range of certain pre-established parameters (e.g., if the IoT nodes measure temperature in a building and send out temperatures of 100°C, the block validation system will detect that it is an anomalous temperature and will not validate that block). Once the block validation system finishes the validation process, it returns a yes or a no to the block validation node. If the answer is no, the block is deleted from the sidechain. If the answer is yes, that block is sent to the smart broker. The block is built in edge computing near the IoT nodes. Once ready to be stored in the blockchain, it is sent to the blockchain's miners' network.

Another of the functions of the block validation node is each time a new IoT device is deployed is to assign a digital identity by using the standard ERC725 blockchain technology. With this standard a smart contract is deployed that implements the ERC725 (Vogelsteller, 2017). In this way, each time a new IoT device is deployed on the IoT network, the block validation node instance a new digital identity for that new node with the information needed to identify it univocally. The steps followed by the validation node block to identify this new IoT devices are as follows:

1. Add a Certifier (i.e., block validation node) and deploy a certifier contract.
2. Add a Protected Contract and deploy a contract called "Listing" with certifier of "IoT network admins". This is the contract which will be limited to interacting to people with verified IoT network accounts.
3. Finds the first wallet ID which is active.
4. Add a new Identity and deploy an identity contract with name "IoT node #1" (or another number).
5. Run Protected Method, and switch the desired Claim Type to "Has rights".
6. Certifier approve the IoT device Identity's claim.
7. You should see that this claim is returned as ClaimValid.

And in this way is how the block validation node approve new IoT devices' identities.

3.2.1.3 Smart broker

The smart broker is the manager of the queues of blocks that reach the miners' network. The block queuing system has a block queuing node on each of the smart controllers. Each of the queuing node blocks will manage a queue of its own blocks. The smart broker sends the blocks to the block queuing system that manages the queue of all smart devices. Block queuing nodes have a queue management system called $M|M|1$. Thus, there is only one block queue, whose capacity is infinite, and only one server (i.e., block queuing system). The discipline is FIFO (First In First Out). Arrivals occur according to a Poisson process of λ reason, where λ is the average number of blocks arriving per unit time and $\frac{1}{\lambda}$ is the average time between the arrival of the blocks.

Theorem 3.1. *Let T be the random variable that represents the time between two consecutive arrivals. Let $t > 0$ and $n(t)$ be the number of arrivals in the system up to the instant t and let λ be the average number of blocks arriving per time unit. Since the increments are independent:*

$$P(T \leq t) = 1 - P(T \geq t) = 1 - P(n(t) = 0) = 1 - e^{-\lambda t} \tag{3.1}$$

then $T = Exp(\lambda)$. Reciprocally, if $T_1, \cdots, T_n$ are independent, where T_i is the time that elapses between the arrivals $(i-1)$-th and i-th, all of them with $Exp(\lambda)$ distribution, then $n(t) = P(\lambda t)$.

Proof. See 3.4.1. □

Then, the times between arrivals of the blocks are distributed exponentially, $Exp(\lambda)$. On the other hand, the times between services will also be distributed exponentially, $Exp(\mu)$, so that μ is the average number of clients the server is capable of serving per unit of time, and $\frac{1}{\mu}$ is the average time of service. To avoid system saturation, it is shown that if $\lambda \geq \mu$ the system is not saturated, thus, the number of blocks in the queue grows indefinitely over time. Therefore, the condition in which the system is not saturated is represented by eq. 3.2.

$$\rho < 1, \text{ where } \rho = \frac{\lambda}{\mu} \tag{3.2}$$

In this paper we are going to assume that block queues are not saturated. When a queue is not saturated, it is also said to be in a stationary state.

ρ parameter is called traffic intensity of the system, since it measures the relationship between the number of arriving blocks and the capacity to process them. Assuming the

system is not saturated, the following formula is deducted from the eq. 3.3 for the p_n odds of there being any blocks in the system, where $n \in \mathbb{N}$.

$$p_n = \rho^n(1-\rho) \tag{3.3}$$

The parameters used to measure the performance of the queues are: 1) Average number of customers (L). To measure the average number of blocks in the system, the eq. 3.4 is used.

$$\begin{aligned} L = &\sum_{j=0}^{\infty} j p_j = \sum_{j=0}^{\infty} j\rho^j(1-\rho) = \\ &= (1-\rho)\sum_{j=0}^{\infty} j\rho^j = (1-\rho)\frac{\rho}{(1-\rho)^2} = \\ &= \frac{\rho}{(1-\rho)} \end{aligned} \tag{3.4}$$

2) Average response time (W). The average response time is the average time a block remains in the system. If we suppose that when a block arrives at the system it is ahead of the other j blocks, the average time taken to exit the system will be $j+1$ times the average time of the service. The calculation of the average response time formula can be found in eq. 3.5.

$$W = \sum_{j=0}^{\infty}(j+1)\frac{1}{\mu}p_j = \sum_{j=0}^{\infty} j\frac{1}{\mu} + \sum_{j=0}^{\infty}\frac{1}{\mu}p_j = \frac{L}{\mu} + \frac{1}{\mu} = \frac{1}{\mu-\lambda} \tag{3.5}$$

where $(j+1)\frac{1}{\mu}$ is the time it takes a block to go through the system if there is j block ahead when it arrives. p_j is the probability that there are j blocks ahead when the block arrives at the system. In this way, the management of queues is mathematically characterized by the block queuing system.

3.2.2 Data analysis layer

The data analysis layer contains the blocks and the blockchain network and all the systems that interact with it directly. In this paper, we propose to add four new blocks to the typical IoT ecosystem used in any architecture. These systems are described in this subsection.

3.2.2.1 Smart management block

The smart management block collects the information from the hashmap control system and depending on the requirement will execute a list of algorithms it contains. These

algorithms will serve to increase the quality of the data, predict future states, increase the fault tolerance of the IoT network, improve maintenance or efficiently control the temperature in a smart building. These algorithms are described in detail in chapter 4.

3.2.2.2 Block validation system

The block validation system is the central node that coordinates all the block validation nodes that are in each of the smart controllers. This node is where the values allowed for the data collected by the IoT nodes are defined. In addition, the block validation node queries the blockchain with the sensor ID and timestamp, to avoid the duplication of data (i.e., the block validation node validates the block if there is no record in the blockchain of the same sensor ID and timestamp in the block.

3.2.2.3 Block queuing system

The block queuing system is the central node of the queuing control of this architecture. Each of the block queuing nodes creates a queue $M|M|1|\infty|FIFO|1$ (i.e., $M|M|1$ abbreviated) with the blocks built by the sidechain. Then, all these queues form a queue network (a queue network is a system where several queues exist and the blocks flow from one queue to another). All queues that coordinate block queuing nodes are directed to the queue that coordinates the block queuing system (probabilistic routing). In addition, the queuing network used in this paper is open (i.e., each block enters the system at a given time, and after passing through one or more queues, exits the system) and cyclic (i.e., a work cannot return to the same queue).

Definition 3.2. *An open queuing network is said to be Jackson's if (if and only if):*

- *There is only one kind of work (i.e., blocks)*
- *Each i node is a queue $.|M|c_i$*
- *On the one hand, routing is probabilistic, where $r_{ij} \geq 0$ is the probability of reaching the node j after leaving the node i. On the other hand r_{i0}, is the probability of leaving the system after leaving node i, where $r_{i0} = 1 - \sum_j r_{ij}$*

Furthermore, the rate of external arrivals to i^{th} node is denoted by γ_i. Moreover, the total number of nodes in the network denotes K.

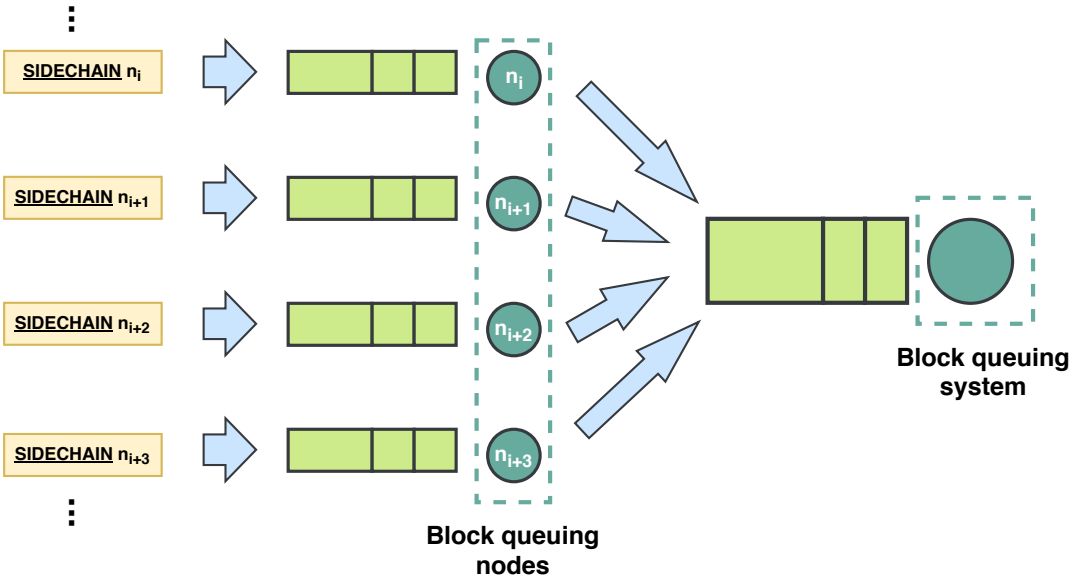

FIG. 3.4: **Block queuing system scheme.** Every single sidechain has its block queuing node. In this way, every block queuing node sends its blocks to the block queuing system.

The queuing network used by the block queuing system is a Jackson network with $r_{ij} = 1$. Thus, all the queues of smart brokers continuously send their blocks to the block queuing system.

Since total input flow to a block queuing node must be equal to the total output flow of the node, the equations of equilibrium must be given by the following equation:

$$\lambda_i = \gamma_i + \sum_{j=1}^{K} \lambda_j r_{ij}, \quad \forall i \in \{1, \cdots, K\} \tag{3.6}$$

The K equations shown in eq. 3.6 form a linear system with a single solution, which we will solve to find the rates of arrival λ_i at the block queuing system. In order to avoid saturation of any of the queuing block queuing system tails, the condition of the eq. 3.7 must be verified.

$$\forall i \in \{1, \cdots, K\}, \quad \rho_i < 1, \text{ where } \rho_i = \frac{\lambda_i}{c_i \mu_i} \tag{3.7}$$

Theorem 3.3. (Jackson's Theorem) *In an open Jackson network of m $M|M|1$ queues where the utilization is $\rho_i < 1$ at every queue, the equilibrium state probability distribution exists and for state $\{(n_1, n_2, \ldots, n_m)\}$ is given by the product of the individual queue equilibrium distributions:*

$$p(n) = \prod_{i=1}^{K} p_i(n_i), \quad \forall n_1, \cdots, n_K \geq 0 \tag{3.8}$$

where $p_i(n_i)$ is the probability of n_i clients in i^{th} node, calculated according to the equations of model $M|M|c$.

Proof. See Medhi [2002], page 228 □

The implications of this theorem are as follows: 1) Overall rate of system outputs (throughput), which is the average number of works leaving the system per unit time, coincides with the number of works entering the system:

$$\lambda_{network} = \sum_{i=1}^{K} \gamma_i \tag{3.9}$$

2) Average number of works in the system, $L_{network}$, which is the sum of the average numbers of works in each of the nodes:

$$L_{network} = \sum_{i=1}^{K} L_i \tag{3.10}$$

3) Average time in the system, $W_{network}$, which is the average time it takes a task to go in and out of the network:

$$W_{network} = \frac{L_{network}}{\lambda_{network}} \tag{3.11}$$

Ratio of visits to node i, V_i, which is the average number of times a work visits node i from the time it enters the network until it leaves:

$$\forall i \in \{1, \cdots, K\},\ V_i = \frac{\lambda_i}{\lambda_{network}} \tag{3.12}$$

An example of a queuing network is shown in Fig. 3.4.

3.2.2.4 Block queuing system: $M|M|1$ queue optimization

We are going to optimize the performance of a $M|M|1$ system in which the server (miners' network) sometimes adjusts the PoW. The server runs blocks until it is empty. It then withdraws and does not offer its service again until there is a Q number of blocks in the queue. The arrival rate is λ, and the service rate is μ, which is $\rho < 1$. The following system costs are defined:

- C_k= cost of monetary units each time the server returns (fixed cost).
- C_h= cost of monetary units per block in the miner's network and time unit (maintenance and mining cost).

The aim is to determine the Q value, enabling the system to operate at minimum cost per time unit. The maintenance cost per unit of time (on average) is $E(N) \cdot C_h$, where $E(N)$ is the number of clients expected in the system. While the fixed cost per time unit is $\frac{C_k}{E(T_0)+T_1}$ where T_0 is the length of a cycle of unemployment and T_1 is the length of a duty cycle. Since the (average) length of a cycle of occupancy and unoccupancy (in steady state) is $E(T_0 + T_1)$. Thus, the function that will be optimized is:

$$C_h E(N) + \frac{C_k}{E(T_0 + T_1)} \tag{3.13}$$

Theorem 3.4. *Let $Q \in \mathbb{N}$ be the number of blocks needed for the miners' network to mine another block again. Let λ be the arrival rate and let μ be the service rate with $\rho < 1$. Let C_k be the monetary units cost each time the server returns and let C_h be the monetary units cost per block in the system and unit time. The optimum value of Q is:*

$$Q^* = \sqrt{\frac{2C_k\lambda(1-\rho)}{C_h}} \tag{3.14}$$

Proof. See 3.4.2. □

3.2.2.5 Block queuing system: IoT data routing algorithm

Here we describe how the adaptive control algorithm works. This algorithm is used by the block queuing system to monitor and control the flow of blocks from the queue network to the miners' network. In Fig. 3.5, the set point (green arrow) with the reference input are the following variables: 1) λ. It is the average number of blocks per unit of time that reach the block queuing system. Each step of the algorithm per time unit, λ is introduced into the control algorithm to update this parameter at each time unit. 2) μ. It is the average number of blocks that the queuing system is able to manage. This parameter enters the flow in each of the steps of the algorithm. However, the adaptive parameters update controls this variable for the optimal performance of the block queuing system. 3) C_k. It is the cost of monetary units each time the server reinitiates its work. 4) C_h. It is the cost of monetary units per block in the miner's network and time unit. Parameters 3 and 4 are calculated considering all the costs associated with each of them.

The adaptive control system controller is composed of the control functions $Q^*(\lambda, \mu, C_k, C_h)$ and $u(z(\lambda, \rho, Q^*))$. The Q^* function estimates the optimal number of blocks for the block queuing system to work on optimizing the energy consumption. We

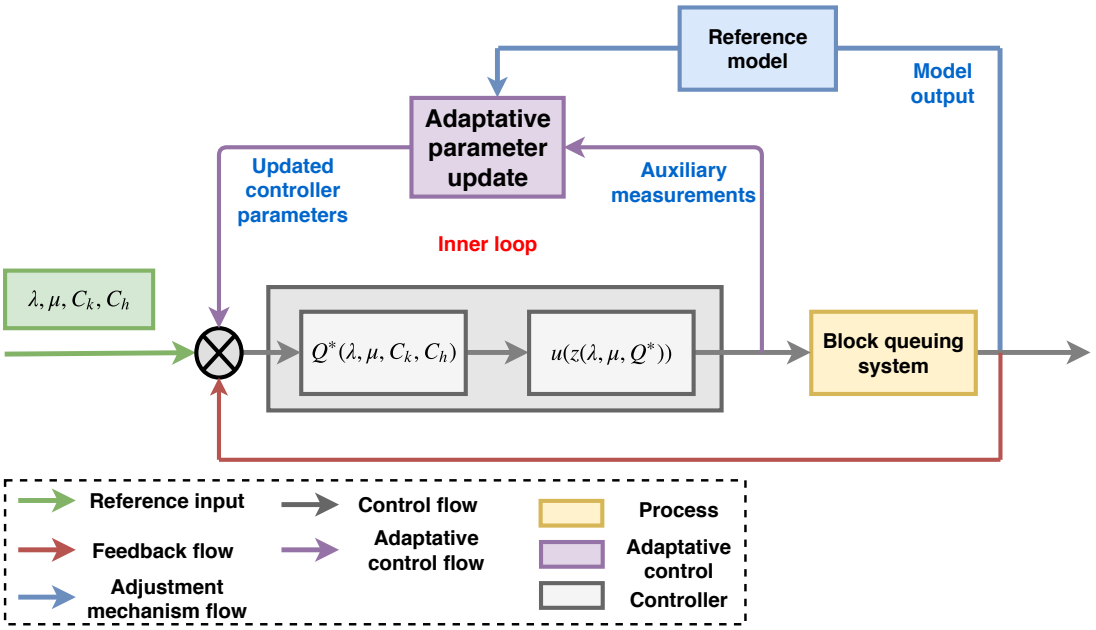

FIG. 3.5: **Auto-adaptive control algorithm**. This algorithm can correct the error in the parameters via the auto-adaptive parameter update function and the reference model (feedback function).

assume that $Q^* = 0$ if $\rho \geq 1$. This is because if $\rho \geq 1$ the Q^* value have a negative square root and in this step of our research we are working with the Real field. Furthermore, the u function decides if μ of the block queuing system changes its value. The Q^* function is defined using the result obtained from the theorem 3.4. We define an auxiliary function z as follows:

$$z(\lambda, \rho, Q^*) = \begin{cases} -1 & if \quad \rho \geq 1 \\ 0 & if \quad \lambda \leq Q^* \ \& \ \rho < 1 \\ 1 & if \quad \lambda > Q^* \ \& \ \rho < 1 \end{cases} \tag{3.15}$$

Then, if $z(\lambda, \rho, Q^*) = 0$, the block control algorithm reduces the number of blocks (μ) that the block queuing system sends to the miners' network. Otherwise, if $z(\lambda, \rho, Q^*) = 1$ the algorithm enables the block queuing system to send blocks into the miners' network. If $z(\lambda, \rho, Q^*) = -1$, the queue is saturated, then the controller does not change the value of μ. The u function is defined by using the auxiliary function z. The value of the parameter $\mu_{controller}$ (μ_c) is given by the following equation:

$$u(z(\lambda, \rho, Q^*)) = \begin{cases} \mu & if \quad u = \pm 1 \\ \frac{1}{2}(Q^* + \lambda) & if \quad u = 0 \\ \mu_a & if \quad \mu \in \text{Inner loop} \end{cases} \tag{3.16}$$

Once the controller has sent the control signal to the block queuing system, it changes the value of μ according to the controller. This is the control flow of the algorithm. Moreover, Fig. 6.9 shows the reference model flow (blue flow), which sends the parameter information μ to the set point which passes it to the controller. In this way, the algorithm receives the reference model information of the process status and confirms whether the queue network collapses or not ($\rho < 1$). The reference model uses the reference function (e) to determine whether the adaptive control algorithm is working properly. Let $\mu_{reference} = \mu_f = w(\rho, Q^*, \lambda)$ be the parameter μ sent by the reference model to the adaptive parameter update. Then, e is defined in the following equation:

$$w(\rho, Q^*, \lambda, \mu_c) = \begin{cases} \frac{(1+\rho^5)}{\pi}(Q^* + \lambda + \mu_c) & if \quad \rho < 1 \\ \frac{1}{5}\lambda\rho^{\frac{7}{3}e}e^{\rho-\pi} & if \quad 1 \leq \rho < \frac{5}{3} \\ \frac{1}{5}\lambda\rho^{2e}e^{\rho-(\frac{e\pi}{2})} & if \quad \rho \geq \frac{5}{3} \end{cases} \tag{3.17}$$

The adaptive parameter update module sends the auxiliary measurements (μ_c) that come out of the controller to the set point, where the value of the μ parameter is changed, so that the controller has the new value of $\mu_{adaptive} = \mu_a = \theta(\rho)$ in the next step of the algorithm. Then, θ is defined as follows:

$$\theta(\rho, \mu_r, \mu_c, Q^*, \lambda) = \begin{cases} \frac{1}{2}(\mu_c + \mu_r) & if \quad \rho < 1 \ \& \ \lambda > Q^* \\ \frac{9}{10\rho^{2\pi e}}(\mu_r + \lambda) & if \quad 1 \leq \rho < \frac{5}{3} \\ \frac{(1+\rho)}{\rho^4}(\mu_r + \lambda) & if \quad \rho \geq \frac{5}{3} \end{cases} \tag{3.18}$$

Hence, the updated controller parameter μ_a is sent to the set point where it is used in one of the steps of the adaptive control algorithm. Also, this updated controller parameter μ_a is sent via the controller to the block queuing system as the new auto-adapt update μ_c. Since the controller system detects that the μ_a comes from the inner loop (i.e., adaptive function), it sends the updated parameter to the block queuing system. Thus, the final signal sent by the controller is $\mu_c = \mu_a$.

3.2.2.6 Hashmap control block

Data are streamed and stored in the blockchain in real time. Once in blockchain, data are sent to the big data layer. In this layer, the data are stored in HDFS. Once data are stored with the Sensor ID and timestamp primary keys, a smart contract is triggered to create a query (e.g., query via Hive), providing access to those data. The smart

contract sends this query, the sensor ID and the timestamp to the "Hashmap control system". This control system stores that data in a hashmap within the blockchain. This minimizes the time required to search through large amounts of data. Although it is not the main objective of this manuscript, we have considered it appropriate to include this system for faster data search. In this way, IoT nodes monitoring and control is ore effective.

3.2.3 Service layer

As described above, the blockchain layer can use the big data layer in real time with bidirectional data streaming. Although the big data layer is essential for the overall functioning of the system, it is not viewed as an important part of this article as it does not represent the main novelty of the work. For this reason, it will not be explained in great detail, although it is necessary to understand that four different blocks can be found in the big data layer: i) Event processing, which is in charge of providing in real time the computing capacity required to respond to the different requests received by the layer; ii) Operational analysis; iii) Data storage, which will generally follow NoSQL models for better performance, although other classic models can also be used; iv) Historical analysis, to extract knowledge from data stored up to the moment of the analysis request by using machine learning algorithms in combination with the business intelligence (BI) of the particular use case.

3.3 Conclusions

The presented IoT architecture allows to the implementation of a system in an smart building. This architecture makes it possible to take energy optimization which allow to achieve energy savings. This reduction in energy consumption can be achieved by, for example, improving control algorithms to setting the temperature to values that do not greatly increase energy consumption. The proposed architecture is able to:

- It is designed in such a way that the system adapts to the technical characteristics of the building, without the need for any additional configuration. Simply by deploying some sensors to obtain the data form the environment.
- This architecture is able to adapt to the site in which the system is deployed.
- It integrates a blockchain which provide security to the system.
- A new routing protocol is designed to improve the send of packages in this architecture.

- It adapts dynamically to the values of the factors with which adjustment decisions are made: the smart management block.

On a technical level, the main advantage of this architecture lies in its modular structure, which allows for the simple addition of new modulate in the data analysis layer. This facilitates the incorporation of artificial intelligence or new control algorithm to improve data quality, robustness of the network or the fault tolerant capacity of the IoT network. This modularity is based on the ability to incorporate new algorithms in the smart management block to improve some of the functions of the IoT network.

3.4 Appendix: Chapter's proofs

In this appendix we shown the proof of the chapter's theorems.

3.4.1 Proof of the Theorem 3.1

Proof.

$$
\begin{aligned}
P(n(t) \leq n) =& P(T_1 + \cdots + T_n + T_{n+1} > t) = \\
&= \int_t^\infty e^{-\lambda t} \frac{\lambda^{n+1} x^n}{n!} dx = \{v = x - t; dv = dx\} = \\
&= \int_0^\infty e^{-\lambda(v+t)} \frac{(v+t)^n \lambda^{n+1}}{n!} dv = \\
&= \int_0^\infty e^{-\lambda(v+t)} \frac{\lambda^{n+1}}{n!} \sum_{i=0}^{\infty} \binom{n}{i} t^i v^{n-i} dv = \\
&= \sum_{i=0}^{\infty} \int_0^\infty e^{-\lambda(v+t)} \frac{\lambda^{n+1} t^i v^{n-i}}{i!(n-i)!} dv = \\
&= \sum_{i=0}^{n} \frac{\lambda^{n+1} t^i}{i!(n-i)!} e^{-\lambda t} \int_0^\infty e^{-\lambda v} v^{n-i} dv = \\
&= \{u = \lambda v; dv = \frac{du}{\lambda}\} = \\
&= \sum_{i=0}^{\infty} \frac{\lambda^{n+1} t^i}{i!(n-i)!} e^{-\lambda t} \int_0^\infty \frac{e^{-u} u^{n-i}}{\lambda^{n-i} \lambda} du = \\
&= \sum_{i=0}^{n} \frac{\lambda^i t^i}{i!} e^{-\lambda t}
\end{aligned}
\tag{3.19}
$$

□

3.4.2 Proof of the Theorem 3.4

Proof. The following notation will be used to demonstrate the theorem:

- $P_n = P(N = n)$
- $P_n(1) = P(N = n)$ and there is a server
- $P_n(0) = P(N = n)$ and there is not a server

It's easy to prove that $P_n = P_n(0) + P_n(1)$.

In the stationary state (from here on, we already assume $\rho < 1$) as a result we obtain the following equations, called equilibrium equations:

Top equation:

$$\begin{aligned}
(n=0) \qquad & \mu P_1(1) = \lambda P_0(0) \\
(n=1) \qquad & \lambda P_0(0) = \lambda P_1(0) \\
(n=2) \qquad & \lambda P_1(0) = \lambda P_2(0) \\
& \vdots \\
(n=Q-1) \qquad & \lambda P_{Q-2}(0) = \lambda P_{Q-1}(0)
\end{aligned} \tag{3.20}$$

Bottom equation:

$$\begin{aligned}
(n=Q) \qquad & \lambda P_{Q-1}(1) + \lambda P_{Q-1}(0) + \\
& + \mu P_{Q+1}(1) = \\
& \lambda P_Q(1) + \mu P_Q(1) \\
(n=1) \qquad & (\lambda+\mu) P_1(1) = \mu P_2(1) \\
(n \geq 2, n \neq Q) \qquad & \lambda P_{n-1}(1) + \mu P_{n+1}(1) = \\
& = (\lambda+\mu) P_n(1)
\end{aligned} \tag{3.21}$$

Merging both systems of equations:

$$\begin{aligned}
\lambda P_0(0) = \lambda P_1(0) = \cdots = \lambda P_{Q-1}(0) \\
\mu P_1(1) = \lambda P_0(0) \\
\lambda P_{Q-1}(1) + \lambda P_{Q-1}(0) + \mu P_{Q+1}(1) = (\lambda+\mu) P_Q(1) \\
\mu P_2(1) = (\lambda+\mu) P_1(1) \\
\lambda P_{n-1}(1) + \mu P_{n+1}(1) = (\lambda+\mu) P_n(1), \;\; n \geq 2, n \neq Q
\end{aligned} \tag{3.22}$$

To solve this system, we will use the probability generator function $G(s)$ associated with the random variable N:

Let $|s| \leq 1$.

$$G(s) = \sum_{n=0}^{+\infty} P_n s^n \quad \left(= \sum_{n=0}^{Q-1} P_n(0) s^n + \sum_{n=1}^{+\infty} P_n(1) s^n \right) \tag{3.23}$$

By defining

$$G_0(s) = \sum_{n=0}^{Q-1} P_n(0) s^n \qquad G_1(s) = \sum_{n=1}^{+\infty} P_n(1) s^n \tag{3.24}$$

Then

$$G(s) = G0(s) + G1(s) \tag{3.25}$$

Moreover

$$G'(s) = \sum_{n=1}^{+\infty} nP_n s^{n-1} \tag{3.26}$$

Thus

$$G(1) = 1 \qquad G'(1) = E(N) \tag{3.27}$$

If we multiply by s^{n+1} the n^{th} equation obtained in the system:

$$\begin{aligned} s\mu P_1(1) + \lambda \sum_{n=2}^{+\infty} P_{n-1}(1)s^{n+1}\lambda P_0(0) + \\ +\mu \sum_{n=1}^{+\infty} P_{n+1}s^{n+1} = \\ = (\lambda+\mu)\sum_{n=1}^{+\infty} P_n(1)s^{n+1} + \lambda P_0(s) \Rightarrow \\ \lambda s^2 G_1(s) + \lambda s^{Q+1} P_0(0) + \mu G_1(s) = \\ = (\lambda+\mu)sG_1(s) + \lambda s P_0(0) \Rightarrow \\ G_1(s) = \frac{\rho s}{1-\rho s}(1 + s + \cdots + s^{Q-1})P_0(0). \end{aligned} \tag{3.28}$$

On the other hand,

$$\begin{aligned} G_0(s) = \sum_{n=0}^{Q-1} s^n P_n(0) = \\ \sum_{n=0}^{Q-1} s^n P_0(0) = (1 + s + \cdots + s^{Q-1})P_0(0) \end{aligned} \tag{3.29}$$

Thus, by evaluating $G(s)$ in $s = 1$ you have to

$$P_0(0) = \frac{1-\rho}{Q} \tag{3.30}$$

So

$$G(s) = G_0(s) + G_1(s) = \cdots = \frac{1 + s + \cdots + s^{Q-1}}{Q}\frac{1-\rho}{1-\rho s} \tag{3.31}$$

Now we should calculate $E(N)$ and $E(T_0 + T_1)$

$$G'(1) = \frac{Q-1}{2} + \frac{\rho}{1-\rho} = E(N) \tag{3.32}$$

$E(T_0)$ = average time for the system to proceed from 0 to Q clients. Since the time between arrivals follows a $Exp(\lambda)$ and this distribution has memory loss, then it is

equivalent to calculating Q times, the average time for a customer to arrive.

$$E(T_0) = QE[e^{\lambda}] = Q\ \frac{1}{\lambda} = \frac{Q}{\lambda} \tag{3.33}$$

$\frac{E(T_0)}{E(T_0+T_1)}$ = failure probability that there are $0.1, ..., Q-1$ clients in the system in steady state and that there is no server

$$\begin{aligned} &= P_0(0) + P_1(0) + \cdots + P_{Q-1}(0) = QP_0(0) \\ E(T_0+T_1) &= \frac{E(T_0)}{QP_0(0)} = \frac{Q\frac{1}{\lambda}}{QP_0(0)} = \frac{1}{\lambda\frac{1-e}{Q}} = \frac{Q}{\lambda(1-\rho)} \end{aligned} \tag{3.34}$$

So

$$\begin{aligned} f(Q) =& C_h E(N) + \frac{C_k}{E(T_0+T_1)} = \\ & C_h(\frac{Q-1}{2} + \frac{\rho}{1-\rho}) + \frac{C_k\lambda(1-\rho)}{Q}, \quad Q = 1, 2, 3, \cdots \end{aligned} \tag{3.35}$$

We look for the value where f reaches its minimum; if Q takes real values

$$\begin{aligned} f'(Q) =& \frac{C_h}{2} - \frac{C_k\lambda(1-\rho)}{Q^2} = 0 \Rightarrow \\ \tilde{Q} =& \sqrt{\frac{2C_k\lambda(1-\rho)}{C_h}} \end{aligned} \tag{3.36}$$

Since

$$f''(\tilde{Q}) = \frac{3C_h}{2}\sqrt{\frac{C_h}{2C_k\lambda(1-\rho)}} > 0 \tag{3.37}$$

Then $\tilde{Q}$ would be local and global minimum.

As Q only takes natural values and the function f is convex, then the optimum solution Q^* will be the one that gives the minimum between $f([Q^*])$ and $f([Q^*]+1)$. Then,

$$Q^* = \sqrt{\frac{2C_k\lambda(1-\rho)}{C_h}} \tag{3.38}$$

As we wanted to prove. □

Chapter 4

Smart management algorithms: Temperature IoT network energy saving optimization

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Chapter 4

Smart management algorithms: Temperature IoT network energy saving optimization

4.1 Introduction

This chapter presents the algorithms developed to optimize energy consumption in smart buildings monitored and controlled by IoT networks. These algorithms are defined as independent modules for their integration in the architecture defined in the chapter 3. But they are designed so that they can be integrated into any architecture or used individually to control the temperature of any smart building.

It is important to note that for this doctoral thesis it has been decided to use temperature as an environmental variable to control, but the algorithms are designed and developed so that any environmental variable can be used. Algorithms presented in this chapter have been developed to solve or improve the gaps that have been found in the literature regarding the optimization of energy consumption through monitoring and control of the IoT network.

First algorithm has been designed to increase the quality of the data collected by IoT networks and to locate the malfunctioning IoT nodes (see subsection 4.2). The main advantages obtained after the design of the data quality algorithm are the self-correction of the wrong data and that it has been able to define a accuracy scale of the devices of an IoT network related to the error found by the data quality algorithm.

Second algorithm that is presented in this chapter is the algorithm for predicting the future precision states of sensors (see subsection 4.3). Using this above mentioned scale and the Markov chains in continuous time, the predictive algorithm of future accuracy states has been designed in order to predict the following accuracy states of the IoT

nodes. In this way, it can be predicted with a certain degree of probability that IoT devices will be in faulty state.

Third algorithm solve one of the gaps found in the literature about the treatment of heterogeneous data in the IoT network. These data do not undergo any type of transformation and are introduced to the control algorithms without taking into account the topology of the networks. In this chapter we present a new method that we have designed to transform heterogeneous data collected by IoT networks into homogeneous data using complex network theory and clustering methods (see subsection 4.4).

Fourth algorithm is a distributed algorithm that has been designed to increase the fault tolerance of an IoT network (see subsection 4.5). This increases the robustness of the IoT network, thus increasing the efficiency in monitoring and control of the IoT network. Algorithms detailed in this chapter optimize the monitoring and control of IoT networks. In addition, the fault tolerance of the IoT networks is increased and by adding these improvements, the energy consumption of the smart buildings is optimised.

Finally, the presented algorithms' contribution to the state of the art is detailed.

The rest of the chapter is organized as follows: section 4.2 shows the detailed description of the IoT data quality algorithm. Future accuracy IoT devices states prediction algorithm is presented in section 4.3, IoT network slicing technique is indeep detailed in section 4.4. In the section 4.5 a new algorithm developed to impreve the robustness in IoT networks is shown. Finally, conclusion are presented in section 4.6.

4.2 IoT data quality algorithm

In this section we provide a mathematical justification of our work. We show how to build the temperature matrix. We also describe the placement of sensors, a mathematical formalization of the designed game and finally its convergence and NE. We apply our algorithm in a case of study and select temperature sensors to test our game.

This section is organized as follows: The construction of the temperature matrix is described in this subsection. The next subsections justify the presented game in mathematical terms. The mathematical formulation of the game and the way in which coalitions are formed. The characteristic function of game is described in next subsection. In the last subsections the paper shows the temperature of the winning coalition, diffuse convergence and the game equilibrium. The mathematical justification is shown below.

4.2.1 Building temperature data matrix.

Let $M \in \mathbb{R}^3$ be a regular known surface on which the temperature will be measured. And let $U \in \mathbb{R}^2$ and $V \in M$ be open sets. Then, there is a local parametrization $\varphi : U \longrightarrow V$, such that (V, φ^{-1}) is a chart from M. Then, a mesh is created in U, and since $\varphi(U) = V$, the mesh can be exported to V. In this way, the points at which the sensors will be placed are obtained as the mesh nodes. Once the sensors are correctly distributed, they begin to collect the temperature of the surface that is being monitored. Then, an area of $M \in \mathbb{R}^3$ and data collection carried out by the sensors $S = \{s_i\}_{i \in \mathbb{N}}$ (i.e. the temperature measurements taken by each sensor) are stored in a matrix in an orderly fashion with respect to the mesh that has been created in $U \in \mathbb{R}^2$ (see Figure 4.1) and $\{p_{i,j}\}_{i,j \in \mathbb{N}}$ are the mesh nodes corresponding with the matrix elements. Each matrix element $s_{i,j}$ correspond to the point $p_{i,j}$ of the ordered mesh in the surface (i.e. $s_{1,1} \rightarrow p_{1,1}$). In the opposite direction is also verified.

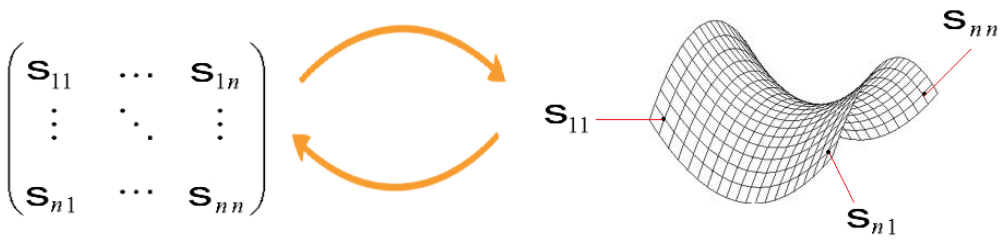

FIG. 4.1: Matrix of sensors and position of the sensors on the surface.

For each sensor location $p_{i,j} \in M$, $T_{p_{i,j}}M$ is the tangent space to M in $p_{i,j}$, and $U \in T_{p_{i,j}}M$ is an open set. Then, we take a sensor $s_{p_{i,j}}$ and a neighbour sensor $s_{p_{i+1,j}}$, and repeat the same process, then an open set $V \in T_{p_{i+1,j}}M$ so that $U \cap V \neq \emptyset$. The U and V charts are compatible, since they meet the compatibility conditions. Now, let's look at the temperatures associated with each sensor in in eq.(4.1):

$$T_{n,n} = \begin{pmatrix} t_{s_{1,1}} & \cdots & t_{s_{1,n}} \\ \vdots & \ddots & \vdots \\ t_{s_{n,1}} & \cdots & t_{s_{n,n}} \end{pmatrix} \tag{4.1}$$

The temperature matrix was interpolated through grade 1 splines, with the new $p_{i,j}$ points created at the intersections of the different openings U_i and V_j (with $U_i \cap V_j$), the interpolation will be done by rows and columns.

Assume there are n + 1 points in row i, $t_0 = p_{s_{i,1}}, \ldots, t_n = p_{s_{i,n}}$, such that $t_0 \leq \cdots \leq t_n$ (i.e. the points are ordered). The intervals $[t_{i-1}, t_i)$ do not intersect each other, so there is no ambiguity in defining the nodes, the grade 1 spline that will be

used to interpolate the new temperatures is as follows:

$$S(x) = \begin{cases} s_0(x) = a_0 x + b_0 & \text{si } x \in [t_0, t_1) \\ \vdots & \\ s_n(x) = a_n x + b_n & \text{si } x \in [t_{n-1}, t_n) \end{cases} \tag{4.2}$$

From this spline function in eq.(4.2), the following transformation can be defined in Eq.(4.3):

$$F_i : T_{n,n} \longrightarrow T^i_{2^i n-1, 2^i n-1} \tag{4.3}$$

such that:

$$\begin{pmatrix} t_{s_{1,1}} & \dots & t_{s_{1,n}} \\ \vdots & \ddots & \vdots \\ t_{s_{n,1}} & \dots & t_{s_{n,n}} \end{pmatrix} \xrightarrow{F_i} \begin{pmatrix} t_{s_{1,1}} & s(t_{s_{1,1}}, t_{s_{1,2}}) & \dots & s(t_{s_{1,n-1}}, t_{s_{1,n}}) & t_{s_{1,n}} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{s_{n,1}} & s(t_{s_{n,1}}, t_{s_{n,2}}) & \dots & s(t_{s_{1,n-1}}, t_{s_{n,n}}) & t_{s_{n,n}} \end{pmatrix} \tag{4.4}$$

Then, the following chain of transformations allows to obtain the new temperature matrices sequentially:

$$T_{n,n} \xrightarrow{F_1} T^1_{2n-1,2n-1} \xrightarrow{F_2} T^2_{4n-1,4n-1} \xrightarrow{F_3} \dots \xrightarrow{F_i} T^i_{2^i n-1, 2^i n-1} \tag{4.5}$$

Therefore, the temperature matrix $T_{n,n}$ is transformed by F_i in the matrix $T^i_{2^i n-1, 2^i n-1}$, where the temperature at the new points is calculated with the interpolation defined by the spline.

GT is a branch of mathematics that is dedicated to the study of conflicting situations that occur when a group of agents with opposing or non-concordant interests, must make their own, individual decisions which affect everyone. Traditionally, this conflict is called game and the player agents. The games almost always present a competition between them. When the rules of the game allow it, players can try to solve the competition through cooperation.

The theory of cooperative games focuses, not on the possible strategies that players can carry out but on the possibilities that coalitions (a coalition is any subset from the total set of players) of these players would have, in a game of N players, there are exactly 2^N coalitions.

4.2.2 Mathematical formulation of the game.

Let $n \geq 2$ denote the number of players in the game, numbered from 1 to n, and let N =$\{1,\ 2,...,n\}$ denote the set of players. A coalition, S, is defined to be a subset of N, $S \subseteq N$, and the set of all coalitions is denoted by $\mathbb{S}$. A cooperative game in N is a function u (characteristic feature of the game) that assigns to each coalition $S_i \subseteq \mathbb{S}$ a real number $u(S_i)$. In addition one has the condition $u(\emptyset) = 0$. In our case, the game will be non-negative (the values of the characteristic function are always positive), monotonous (if more players are added to the coalition the value of the expected characteristic function does not change), simple and 0-normalized (players are obliged to cooperate with each other since individually they will obtain zero benefit).

In our case, the set of players is the set of ordered sensors S and the characteristic function u is defined as:

$$u : 2^n \longrightarrow \{0, 1\} \tag{4.6}$$

such that, for each coalition of sensors, u = 1 or 0 if that particular coalition can vote or not respectively (see Eq.(4.6, 4.7)).

$$\mathbb{S} \ni S_i \longrightarrow u(S_i) = \{0, 1\} \in \mathbb{R} \tag{4.7}$$

4.2.2.1 Cooperative sensor coalitions

Cooperative games are defined by the fact that players can cooperate with each other in order to achieve a mutual benefit. Once the players have agreed to cooperate among themselves, a coalition should be formed. A coalition can be formed by any number of players. In our game, a single sensor cannot determine if its temperature is correct. Therefore, sensors are forced to cooperate in order to evaluate whether the temperature of the central sensor is correct in relation to their neighbourhood. Sensors work together in coalitions in order to cooperate for a mutual benefit (i.e., to verify that their temperatures are correct and to self-correct them if this is not the case). Sensors will work together by majority rule-based voting among the coalition members in the game presented in this paper. But not all sensors can form coalitions, so it is necessary to establish limitations for the allowed coalitions that could potentially be formed regarding the respective central sensor.

The possible coalitions that the sensors will form, will be limited by their position, that is, the coalitions can only be formed by neighbouring sensors. Let's consider the matrix of the sensors and a pair of sensors $s_{i,j}$ y $s_{k,m}$ will be in the same neighbourhood if and

only if:

$$\| (i-k)^2 - (j-m)^2 \| \leq 1 \tag{4.8}$$

that is, if each sensor to which the game is applied, is the center of a Von Neumann neighbourhood, its neighbours are those lying within a Manhattan distance (in the matrix) equal to one. In addition, the following conditions have to be fulfilled by the allowed coalitions:

1. Coalition sensors have to be in the same neighborhood as defined in Eq.4.8.
2. Coalitions cannot be formed by a single sensor.

Assuming, that the game is applied to sensor $s_{i,j}$, the allowed coalitions are:

- $S_1 = \{s_{i,j}, s_{i+1,j}\}$
- $S_2 = \{s_{i,j}, s_{i-1,j}\}$
- $S_3 = \{s_{i,j}, s_{i,j-1}\}$
- $S_4 = \{s_{i,j}, s_{i,j+1}\}$
- $S_5 = \{s_{i,j}, s_{i,j+1}, s_{i+1,j}\}$
- $S_6 = \{s_{i,j}, s_{i,j+1}, s_{i-1,j}\}$
- $S_7 = \{s_{i,j}, s_{i-1,j}, s_{i,j-1}\}$
- $S_8 = \{s_{i,j}, s_{i-1,j}, s_{i,j-1}\}$
- $S_9 = \{s_{i,j}, s_{i,j-1}, s_{i+1,j}\}$
- $S_{10} = \{s_{i,j}, s_{i,j-1}, s_{i,j+1}\}$
- $S_{11} = \{s_{i,j}, s_{i-1,j}, s_{i+1,j}\}$
- $S_{12} = \{s_{i,j}, s_{i-1,j}, s_{i+1,j}, s_{i,j+1}\}$
- $S_{13} = \{s_{i,j}, s_{i-1,j}, s_{i+1,j}, s_{i,j-1}\}$
- $S_{14} = \{s_{i,j}, s_{i,j+1}, s_{i,j-1}, s_{i+1,j}\}$
- $S_{15} = \{s_{i,j}, s_{i,j+1}, s_{i,j-1}, s_{i-1,j}\}$
- $S_{16} = \{s_{i,j}, s_{i,j+1}, s_{i,j-1}, s_{i-1,j}, s_{i+1,j}\}$

Therefore, the set of allowed coalitions is: $\mathbb{S} = \{s_i\}_{1 \leq i \leq 16}$. Thus, the game can be expressed as a majority game that is weighted according to the formed coalitions in the following way:

$$u \equiv \{\frac{n}{2} + 1; s_1, \ldots, s_n\} \text{ where } s_i \text{ are all neighbours and } s_i \in \mathbb{S} \tag{4.9}$$

Figure 4.2 illustrates a flowchart showing the steps involved in deciding if a coalition is allowed or not allowed.

4.2.2.2 A characteristic function to find *cooperative* temperatures.

In the proposed game, we want the neighbourhood coalitions to democratically decide the temperature of the main sensor. To do this, they will form coalitions that will decide on the final temperature of the sensor, which will be determined by whether they can

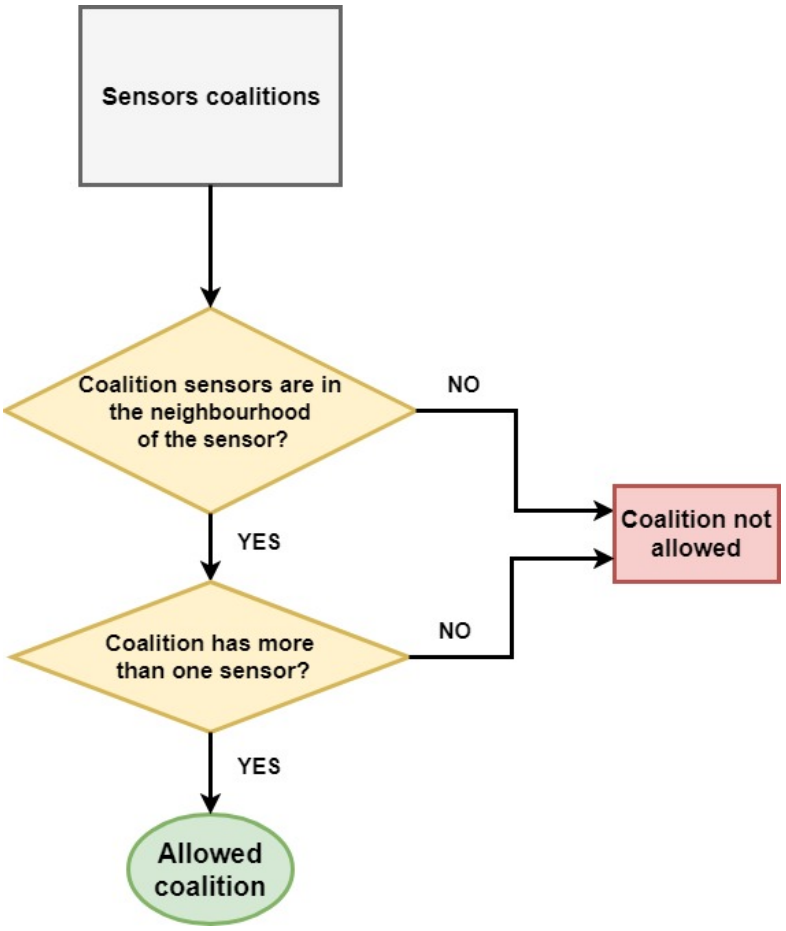

FIG. 4.2: Allowed coalition decision making flowchart.

vote or not in the process. From the characteristic function defined in eq.(4.6), if the value is 1(0), the coalition can vote (not vote) respectively. s_i is the main sensor with its associated temperature t_{s_i}, the characteristic function is built in the following way:

1. First, the average temperature of all the sensors is calculated:

$$T_{s_i}^k = \frac{1}{V} \sum_{i}^{V} t_{s_i} \tag{4.10}$$

   here $T_{s_i}^1$ represents the average temperature of the sensors' neighbourhood s_i (including it) in the first iteration of the game and V is the number of neighbours in the coalition.

2. The next step is to compute an absolute value for the temperature difference between the temperatures of each sensor and the average temperature:

$$\overline{T}_{s_i}^k = \left(\frac{1}{V} \sum_{i}^{V} \mid t_{s_i} - T_{s_i}^k \mid^2 \right)^{\frac{1}{2}} \tag{4.11}$$

3. Using the differences in temperature values with regards to the average temperature $\overline{T}_{s_i}^k$ (see eq.(4.11)) a confidence interval is created and defined as follows:

$$I_{s_i}^k = \left(T_{s_i}^k \pm t_{(V-1,\frac{\alpha}{2})} \frac{\overline{T}_{s_i}^k}{\sqrt{V}} \right) \tag{4.12}$$

   in Eq.(4.12) we use the Student's-t distribution with an error of 1%.

4. In this step we use a hypothesis test. If the temperature of the sensor lies in the interval $I_{s_i}^k$, it belongs to the voting coalition, otherwise, it is not in the voting coalition:

$$u^k(s_1, \ldots, s_n) = \begin{cases} 1 & \text{if } t_{s_i} \in I_{s_i}^k \\ 0 & \text{if } t_{s_i} \notin I_{s_i}^k \end{cases} \tag{4.13}$$

5. The characteristic function will repeat this process iteratively (k is the number of the iteration) until all the sensors in that iteration belong to the voting coalition. In each iteration k, the following payoff vector of the coalition S_j (with $1 \leq j \leq n$ where n is the number of sensors in the coalition) in the step k ($PV(S_j^k)$) is available:

$$PV(S_j^k) = (u^k(s_1), \ldots, u^k(s_n)) \text{ where } \sum_{i}^{n} u^k(s_i) \leq n \tag{4.14}$$

   The stop condition of the game iterations is $PV(S_j^k) = PV(S_j^{k+1})$ the process end. That is, let $PV(S_j^k) = (u^k(s_1), \ldots, u^k(s_n))$ and let $PV(S_j^{k+1}) = (u^{k+1}(s_1), \ldots, u^{k+1}(s_n))$. The iteration process ends when both payoff vectors contain the same elements. This process is shown in the following equation:

$$\begin{cases} u^k(s_1) = u^{k+1}(s_1) \\ \vdots \\ u^k(s_n) = u^{k+1}(s_n) \end{cases} \tag{4.15}$$

   Then the game can find the solution that is defined in the following subsection.

4.2.2.3 Solution concept of the cooperative game

Once the characteristic function has been applied to all sensors involved in this step of the game a payoff vector in the step k is available (see eq.4.14). Since the proposed game is a cooperative game, the solution concept is a coalition of players that we have called game equilibrium (GE). The GE of the proposed game is defined as the minimal coalition with more than half of the votes cast. Below, we summarize the conditions that must be met by the winning coalition for the game to reach the GE. Let n be the number of players involved in this step of the game. Winning coalition must satisfy the following conditions:

1. Sum of the elements of the coalition PV must be higher than half plus 1 of the votes cast:
$$\sum_{i}^{n} u^k(s_i) \geq \frac{n}{2} + 1 \tag{4.16}$$
2. The coalition is maximal (i.e., coalition with the greatest number of elements, different from 0, in its payoff vector $PV(S_j^k)$).

Therefore, the solution to the proposed game is the coalition that verifies both conditions from among all possible coalitions that are formed at each step k of the game.

4.2.2.4 Temperatures of the winning coalition.

Once the characteristic function decides which is the winning coalition, it is possible to calculate the temperature of the main sensor. Let $\{s_1, \ldots, s_j\}$ be the winning coalition's sensors and $\{t_{s_1}, \ldots, t_{s_j}\}$ be their associated temperature.
The temperature that the game has voted to be the main sensor's temperature (MST) is calculated as follows:

$$MST = \max_{j \in |S_{winner}|} \{j * t_{s_i}\}_{s_{i \in S_{winner}}} \tag{4.17}$$

where $|\mathbb{S}|$ is the number of elements in the winning coalition. Therefore, the MST will be the maximum temperature that has the highest relative frequency. In case of a draw, it is resolved by the Lagrange criterion.

4.2.2.5 Diffuse convergence.

In each game iteration, there is a matrix with temperature (see Eq.4.1). Hence we define a sequence of arrays $\{M_n\}_{n \in \mathbb{N}}$ where the M_i element corresponds to the temperature

matrix in step i of the game. Therefore, it can be said that the sequence of matrices is convergent if:

$$\forall \epsilon > 0, \text{ there is } N \in \mathbb{N} \text{ such that } |M_{i-1} - M_i| \leq \epsilon \ \forall i \in \mathbb{N} \ , \forall i \geq N. \tag{4.18}$$

That is, if the element $m_{n,m}^{i-1} \in M_{i-1}$ and the element $m_{n,m}^{i} \in M_i$ are set and the convergence criterion is applied, we have:

$$\begin{aligned} \forall \epsilon_{n,m} > 0 \text{ there is } N \in \mathbb{N} \text{ such that } |m_{n,m}^{i-1} - m_{n,m}^{i}| \leq \epsilon_{n,m} \\ \forall i \in \mathbb{N} \ , \forall i \geq N \text{ and } m_{n,m}^{i-1} \in M_{i-1}, m_{n,m}^{i} \in M_i \end{aligned} \tag{4.19}$$

Therefore, by applying the criterion of convergence in Eq.(4.18) to all the elements, a new matrix is obtained with the temperature differences between the temperatures obtained in previous and in the next step of the game.

$$\begin{pmatrix} |m_{1,1}^{i-1} - m_{1,1}^{i}| & \dots & |m_{1,m}^{i-1} - m_{1,m}^{i}| \\ \vdots & \ddots & \vdots \\ |m_{n,1}^{i-1} - m_{n,1}^{i}| & \dots & |m_{n,m}^{i-1} - m_{n,m}^{i}| \end{pmatrix} \tag{4.20}$$

For the succession of matrices to be convergent, each of the sequences of elements that are formed with the $|m_{n,m}^{i-1} - m_{n,m}^{i}|$ must be less than the fixed $\epsilon > 0$. In this work, it is established that $\epsilon = 0.01$.

Furthermore, the game must incorporate fuzzy logic, since it makes decisions about the temperature of the main sensor. To this end, we introduce the concept of Fuzzy Matrix Convergence (FMC), starting from the previous definition of matrix convergence, we will say that a matrix is FMC if at least 80 % of the elements of the matrix are convergent. This allows us to improve the efficiency with which the temperature matrix reaches convergence. In addition, this allows for the convergence of the game to be quite fast in its respective iterations. When the matrix reaches the FMC, temperatures have been self-corrected in a distributed and self-organized way.

4.2.3 Distributed and self-organized justification.

One of the main goals of our work is that the game is distributed and self-organized. It is distributed because the game can run without the need for a central node (i.e. when game is applied to every single node, this node is the *central node* in this step, but in next step other nodes will be the *central node*). On the other hand, the game is self-organized because in each step the game uses the output of the previous step as the

input of the next step. In this way, game players (i.e. IoT nodes) interact in every game step between them to form temperature clusters with their neighbouring sensors.

4.2.4 Algorithm architecture

Fig.4.3 presents the architecture of the algorithm proposed in this paper. The architecture has several layers. First is the WSN where the sensors that collect the smart building temperatures are located. In the next layer is the proposed algorithm. This algorithm has 3 steps: 1) in the first step the allowed coalitions of the sensors are formed. 2) In the second step the game we have created for this algorithm is applied and the winning coalition is voted by majority rule. 3) In the last step, the temperature of the winning coalition is calculated, which is the cooperative temperature of the sensor to which the game is applied.

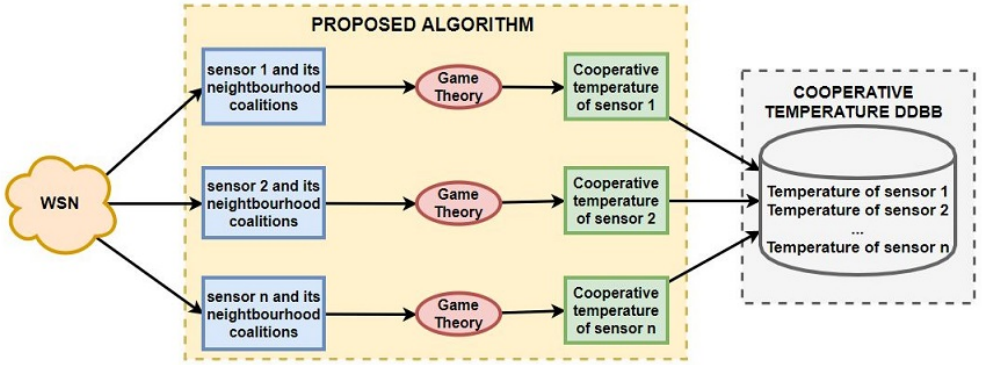

FIG. 4.3: Architecture of the proposed algorithm.

4.3 Future accuracy IoT devices states prediction algorithm

In this section, we propose a new feedback control algorithm for predictive maintenance and improve the monitoring and control of the IoT networks.

4.3.1 Initial accuracy state

Initially, it is necessary to define a scale of accuracy degradation expressed in percentages according to the data obtained in the execution of the algorithm that we had developed in a previous research Casado-Vara et al. [2019a]. This scale will be the discussion universe of the random variable X_n that defines the current state of precision of the system related to the error of the sensors. Therefore, of the sensors' possible states are

$X_n = \{A = high\ accuracy,\ B = \ accurate,\ C = low\ accuracy,\ F = failure\}$. Below, table 4.1 has the selection made for each variable.

X_n	Sensor accuracy state	Error (%)
A	High accuracy	e $\leq$ 10
B	Accurate	10 ¡ e $\leq$ 20
C	Low accuracy	20¡ e $\leq$ 35
F	Failure	e $\geq$ 35

TAB. 4.1: Accuracy state of sensors.

Let $T_i^{(t)}$ be the matrix of initial temperatures collected by the WSN, and let $T_f^{(t)}$ be the final temperatures after applying the data quality algorithm. Then, the accuracy error matrix of the sensors according to the data quality algorithm given by the following equation:

$$T_e^{(t)} = |T_f^{(t)} - T_i^{(t)}| \tag{4.21}$$

where the coefficients e_{ij} of the matrix $T_e^{(t)}$ are the differences between the initial and final temperature in absolute value for each sensor.

Given the $T_e^{(t)}$ matrix, we now apply the error correction given by the allowed error margin ϵ, and adjust the error matrix:

$$T_\epsilon^{(t)} = |T_e^{(t)} - Id * \epsilon| \tag{4.22}$$

Now, let's centralize these measures to calculate the states of the sensors. To this end, we calculate the average of the elements of the array m_ϵ and the maximum of the array $T_\epsilon^{(t)}$ that we call max_ϵ. Therefore, the centralizing measure is defined as:

$$\delta = m_\epsilon + max_\epsilon \tag{4.23}$$

this measure is applied to the matrix $T_\epsilon^{(t)}$ to calculate the percentages associated with each error and therefore calculate the states of each sensor:

$$T_\delta^{(t)} = \begin{pmatrix} t_{1,1}^\delta = \frac{(t_{1,1}*100)}{\delta} & \cdots & t_{1,n}^\delta = \frac{(t_{1,n}*100)}{\delta} \\ \vdots & \ddots & \vdots \\ t_{n,1}^\delta = \frac{(t_{n,1}*100)}{\delta} & \cdots & t_{n,n}^\delta = \frac{(t_{n,n}*100)}{\delta} \end{pmatrix} \tag{4.24}$$

Then, one can define the following function:

$$g^{(t)} : M_{n,n}(\mathbb{R}) \longrightarrow M_{n,n}(\{X_n\}) = T^{g(t)} \tag{4.25}$$

defined as follows:

$$g^{(t)}(t_{i,j}^{\delta}) = \begin{cases} A & if \quad t_{i,j}^{\delta} \leq 10\% \\ B & if \quad 10\% < t_{i,j}^{\delta} \leq 20\% \\ C & if \quad 20\% < t_{i,j}^{\delta} \leq 35\% \\ F & if \quad t_{i,j}^{\delta} \geq 35\% \end{cases} \tag{4.26}$$

where $t_{i,j} \in T_{\delta}^{(t)}$, and let $T^{g(t)}$ be the matrix with the accuracy states of the sensors at time t.

4.3.2 Transition matrix

Let λ_A be the time the sensor is in state A (exponential distribution). λ_B and λ_C are defined in a similar way. And let ξ_A be the time the sensor remains in state A. Let μ_A (μ_B, μ_C) be the probability that a sensor be in state A (B,C) at time t moves to state F in the time interval $(t, \Delta t + t)$. Thus, if the sensor was in state A at time t_i , the probability of the sensor remaining in state A at time t_{i+1} is given by the following equation:

$$P(\xi_A > t + \Delta t | \xi_A > t) = \frac{e^{-\lambda_A(t+\Delta t)}}{e^{-\lambda_A t}} = e^{-\lambda_A \Delta t} = 1 - \lambda_A \Delta t + o(\Delta t) = p_{AA} \tag{4.27}$$

Similarly, the probability that a sensor that is in state A at the beginning, ends up being in state B, is given by the following equation

$$P(\xi_B > t+\Delta t | \xi_A > t) = 1-((1-\lambda_A \Delta t+o(\Delta t))-(\mu_A \Delta t+o(\Delta t))) = (\lambda_A-\mu_A)\Delta t+o(\Delta t) = p_{AB} \tag{4.28}$$

In this way, we can build the transition matrix between t and $t + \Delta t$, where the coefficients of the transition matrix are the probabilities of the sensors' switching states (e.g., p_{AF} is the probability that a sensor that begins in state A and ends up being in state F in the interval $(t, \Delta t + t)$).

In this way, the transition matrix $P(t)$ is built:

$$P(t) = \begin{pmatrix} P(\xi_A > t + \Delta t | \xi_A > t) = p_{AA} & \dots & tP(\xi_F > t + \Delta t | \xi_A > t) = p_{AF} \\ \vdots & \ddots & \vdots \\ P(\xi_A > t + \Delta t | \xi_F > t) = p_{FA} & \dots & P(\xi_F > t + \Delta t | \xi_F > t) = p_{FF} \end{pmatrix} \tag{4.29}$$

4.3.2.1 Predictive control algorithm

Here we describe how the control algorithm works. This algorithm is used by the sensor control system to monitor and control the accuracy of the sensors. In Fig. 4.4, the set point (green arrow) with the reference inputs are the following variables: 1) The accuracy error matrix, T_e (see eq. 4.21). This matrix has the precision errors of the mesh of sensors. Each step of the algorithm at every time t, this matrix is introduced to update the data of the algorithm. 2) The allowed error ϵ. This parameter enters the flow in each of the steps of the algorithm.

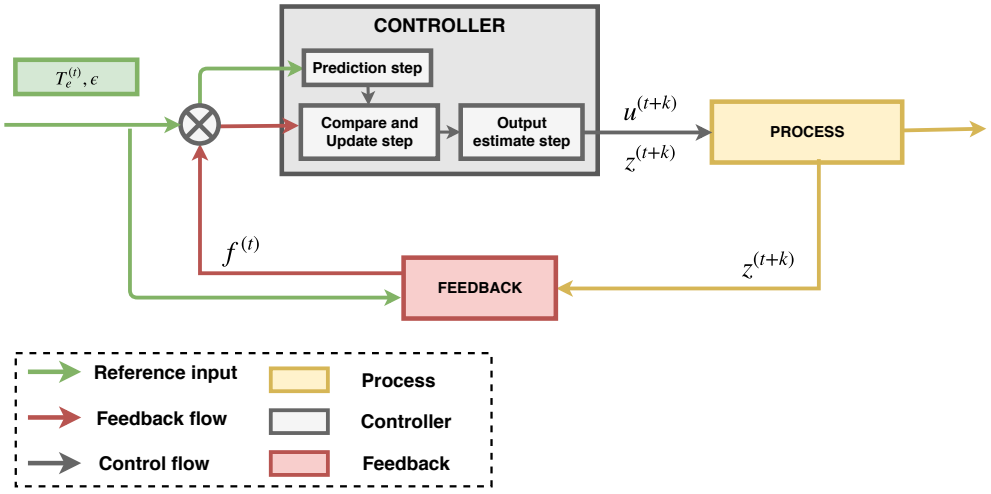

FIG. 4.4: This algorithm predicts the accuracy state of the sensors via the feedback control algorithm in the time interval $(t, t + \Delta t)$.

4.3.2.2 Controller

The first action performed in the controller is the prediction step. In this stage of the algorithm, the transition matrix of the developed model is used. (see eq.(4.29)). Let $z^{(t)} : T^{g(t)} \longrightarrow z^{(t)}(T^{g(t)}) = T^{z(t+k)}$ be the prediction function of accuracy states (i.e., Prediction step) for each time t and let $t+k$ where $k \in \{1, 2, \cdots\}$ be the predicted time. Given $t^{\delta}_{i,j} \in T^{\delta}$, the controller function u is defined as follows:

$$z_{ij}^{(t+k)}(t^{g}_{i,j}) = max\{\mathbb{P}_{t^{g(t+k)}_{i,j}A}, \mathbb{P}_{t^{g(t+k)}_{i,j}B}, \mathbb{P}_{t^{g(t+k)}_{i,j}C}, \mathbb{P}_{t^{g(t+k)}_{i,j}F}\} \tag{4.30}$$

Let $z^{(t)}(T^{g}) = T^{z(t+k)}$ be the matrix of the states of accuracy given by the prediction function. The output of this function is the accuracy state of the sensors at time t.

The next step of the algorithm is to compare the measurements with the feedback function in order to update them. Let $x^{(t)} : T^{z(t)} x T^{f(t-k)} \longrightarrow x^{(t)}(T^{z(t)}) = T^{x(t)}$ be the comparison function defined by the following numerical values $\{A = 1, B = 2, C =$

$3, F = 4\}$ as follows:

$$x^{(t)}(t_{i,j}^{z(t)}, t_{i,j}^{f(t-k)}) = w_{x_1(t)} t_{i,j}^{z(t)} + w_{x_2(t)} t_{i,j}^{f(t-k)} \tag{4.31}$$

where $w_{x_n(t)}$ with $n \in \{1, 2\}$ are the given weights for each of the coordinates of the function x.

Let $y^{(t)} : T^{x(t)} \longrightarrow y^{(t)}(T^{x(t)}) = T^{y(t)}$ be the update function defined as follows:

$$y^{(t)}(T^{z(t)}, T^{f(t)}) = \begin{cases} 1 & if \quad 0 \leq t_{i,j}^{h(t)} \leq 1.5 \\ 2 & if \quad 1.5 < t_{i,j}^{h(t)} \leq 2.5 \\ 3 & if \quad 2.5 < t_{i,j}^{h(t)} \leq 3.5 \\ 4 & if \quad t_{i,j}^{h(t)} \geq 3.5 \end{cases} \tag{4.32}$$

The update function refreshes the accuracy states of the prediction function with the results obtained from the comparison function.

Let $u : T^{y(t)} \longrightarrow u^{(t)}(T^{y(t)}) = T^{u(t)}$ the controller function (i.e., output estimate step) and let $T^{u(t)}$ the system controller matrix at time t. Then, this function finds sensors that are in a state of failure (F). In this way, the system creates a virtual sensor to maintains the monitoring of the system. In addition, it will send a request to the service staff to replace the failure sensor. Given $t_{i,j}^{y(t)} \in T^{y(t)}$, u is defined as follows:

$$u(t_{i,j}^{y(t)}) = \begin{cases} 1 & if \quad t_{i,j}^{y(t)} = F \\ -1 & if \quad t_{i,j}^{y(t)} \neq F \end{cases} \tag{4.33}$$

thus, if $u(y^{(t)}) = 1$, the system creates a virtual sensor in the position (i, j) and asks for maintenance.

4.3.2.3 Feedback

Let $h^{(t)} : T^{g(t)} x T^{g(t+k)} x T^{z(t+k)} \longrightarrow h^{(t)}(T^{z(t+k)}) = T^{h(t)}$ be the auxiliary feedback function. Given $k \in \{1, 2, \cdots\}$ and the accuracy states in numerical values are $\{A = 1, B = 2, C = 3, F = 4\}$, h is defined as follows:

$$h^{(t)}(t_{i,j}^{g(t)}, t_{i,j}^{g(t+k)}, t_{i,j}^{z(t+k)}) = w_{h_1(t)} t_{i,j}^{g(t)} + w_{h_2(t)} t_{i,j}^{g(t+k)} + w_{h_3(t)} t_{i,j}^{z(t+k)} \tag{4.34}$$

where $w_{h_n(t)}$ with $n \in \{1, 2, 3\}$ are the given weights for each of the coordinates of the function h.

Let $f^{(t)} : T^{h(t)} \longrightarrow f^{(t)}(T^{h(t)}) = T^{f(t)}$ be the feedback function defined as follows:

$$f^{(t)}(T^{h(t)}) = \begin{cases} A & if \quad 0 \leq t_{i,j}^{h(t)} \leq 1.5 \\ B & if \quad 1.5 < t_{i,j}^{h(t)} \leq 2.5 \\ C & if \quad 2.5 < t_{i,j}^{h(t)} \leq 3.5 \\ F & if \quad t_{i,j}^{h(t)} \geq 3.5 \end{cases} \tag{4.35}$$

The feedback function returns the accuracy state of the sensor (i, j). In this way, it is verified that the controller is working correctly and that virtual sensors are not created for the repair of sensors that are working properly.

4.3.2.4 Process.

The process matrix $T^{p(t)}$ shows when sensors need maintenance. The process matrix is defined as follows:

$$T^{p(t)} = T^{u(t-1)} + T^{u(t)} \tag{4.36}$$

Thus, when the coefficient of the matrix correspond to a particular sensor, it means that it has to be replaced $t_{(i,j)}^{p(t)} \geq 0.5\% t_{max}$ time periods with $t_{(i,j)}^{p(t)} \in T^{p(t)}$ (i.e., assuming that $t_{max} = 5$ years, then a sensor has to be replaced if $t_{(i,j)}^{p(t)} \geq 9$ days).

Then, the controller function sends a signal to the process which sends back the matrix of final virtual temperatures at time t (i.e., $T_{vf}^{(t)}$). When the controller sends the signal that a sensor is in failure state, the process creates a virtual sensor in that position and simulates the temperature so that the monitoring and control of the building does not lose efficiency. Let $\{T_f^{(t)}\}_{t \geq 0}$ be the matrix succession with the final temperatures at time t given by the algorithm described in Casado *et al.* Casado-Vara et al. [2019a] (see appendix A). Moreover, let $VS_{i,j}^{(t)}$ be the virtual sensor in the position (i, j) at time t. Then, the temperature of the $t_{i,j}^{v}$ is provided by the temperature $t_{i,j} \in T_f^{(t)}$.

4.4 IoT network slicing

In this section, the solution that we propose to address the above problem is described. The inaccuracy problem arises from the application of temperature control algorithms to IoT nodes with heterogeneous data. To counteract this problem we propose the

combination of several mathematical and artificial intelligence techniques with which we have developed an intelligent and self-adaptive model that allows for the use of temperature control algorithms in all types of IoT networks. The operation of this model begins with the collection of data by the IoT nodes. These data are usually heterogeneous (i.e., the temperatures collected by the IoT network in a smart building are very different depending on the area of the smart building where they are collected). The proposed model encompasses the following techniques or algorithms:

1. First, a graph is constructed in which the nodes will be the IoT nodes and the edges of the graph will be the doors or corridors that join those IoT nodes. Since a graph has been built with the IoT network, a complex network can then be built with this graph.
2. Now we apply clustering algorithms using the temperature data collected by the nodes of the complex network (i.e., IoT nodes). In this way, heterogeneous data are separated into homogeneous clusters.
3. Next, a multiplex is built in which each of the layers represents one cluster.
4. Multiplex layers that have unconnected networks will use virtual nodes to build a related network. In this way the algorithms can be applied correctly in the following stages.
5. The control algorithms required to be used will be applied depending on the purpose on homogeneous data in each of the layers.
6. Each layer is projected on to the complex network and the control signal is sent to each one of the actuators assigned to the IoT nodes.

In this section, we describe all the techniques that are required for the development of this model. A flowchart summarizing this model is presented in Fig. 6.16.

4.4.1 Graph design module

The graph is constructed from the topology of the IoT nodes in the intelligent building. In this way, a graph is formed where the vertices of the graph are the IoT nodes and the edges of the graph are the physical connections between the rooms where the IoT nodes are located (i.e. there is no obstacle on the way from one node to another). That is, if between the rooms in which the IoT nodes are located there is a door and a corridor, then there is an edge in the graph. The graph represents the heat transfer between the physically connected rooms. An illustrative example can be found in Fig. 4.6. This is

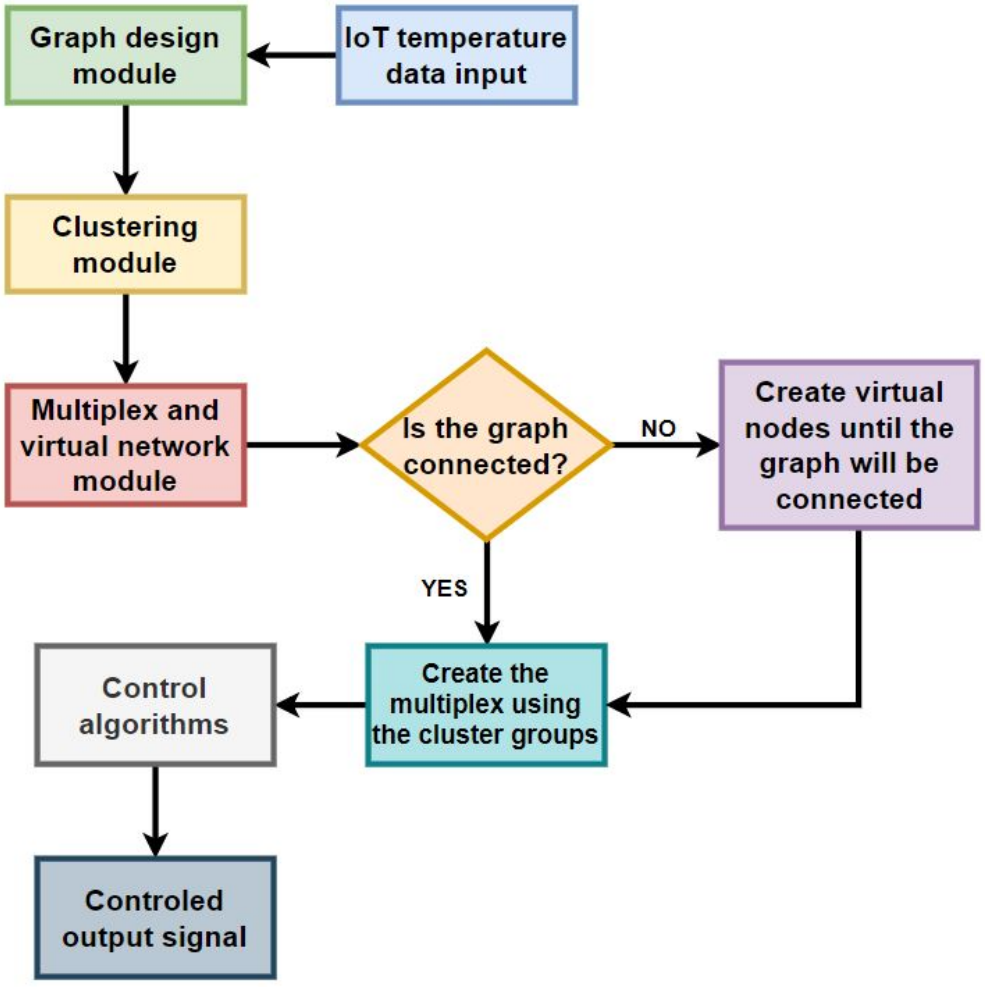

FIG. 4.5: Proposed model flowchart.

how we build a graph from the IoT network. Now let's consider this graph as a complex network, also we consider non-directed graphs. Then, we create the *Laplacian matrix* of the graph as follows:

$$A = \begin{cases} 1 & \text{if } (i,j) \in E \\ t_i & \text{if node j = i } (t_i = \text{temperature of the i}th \text{ node}) \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

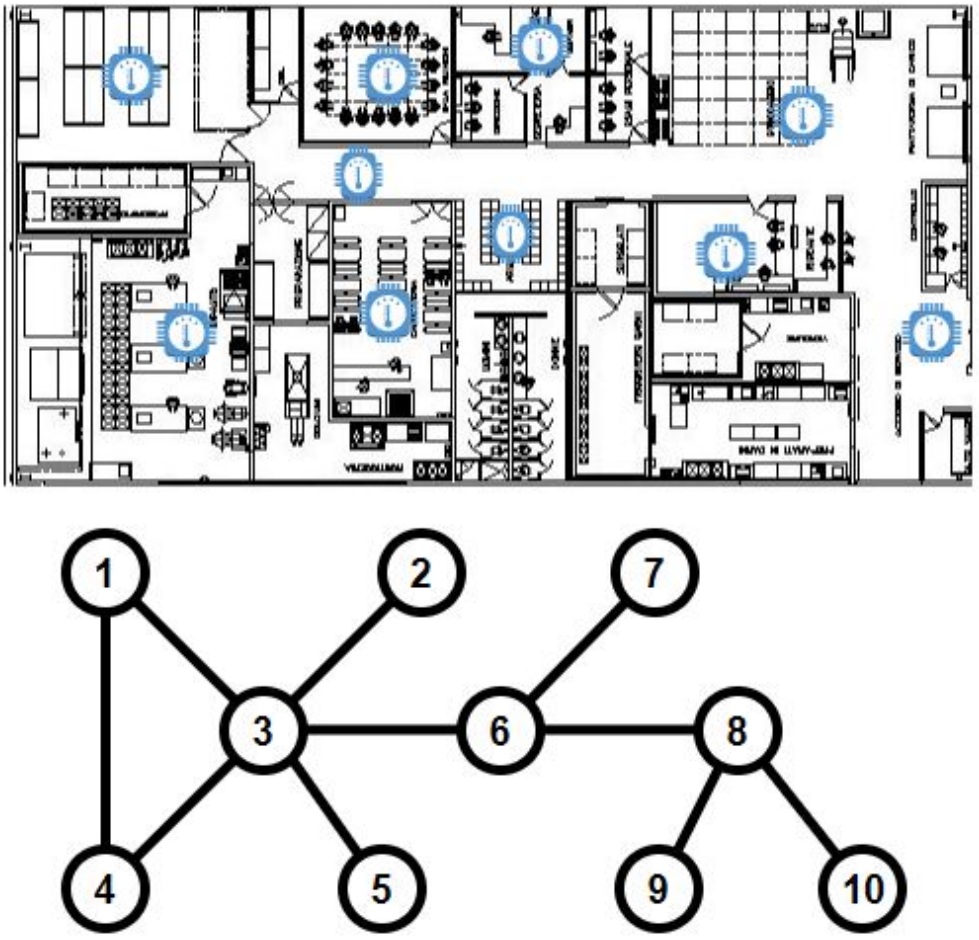

FIG. 4.6: Illustrative example of the construction of a graph based on the position of the IoT nodes on a map.

where E is the set of edge in the graph. For the example proposed in the figure, the *Laplacian matrix* is:

$$A = \begin{pmatrix} t_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & t_3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & t_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & t_6 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & t_7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & t_8 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & t_{10} \end{pmatrix} \tag{4.38}$$

4.4.2 Clustering module

The temperature data collected by the IoT network of the smart building are normally heterogeneous. By applying a clustering algorithm, we will separate these data into groups of homogeneous temperatures. Gaussian Mixture Models (GMMs) give us more flexibility than K-Means. With GMMs we assume that the data points are Gaussian distributed; this is a less restrictive assumption than the k-means algorithm uses the mean to form circular clusters. That way, we have two parameters to describe the shape of the clusters: the mean and the standard deviation. With an appropriate amount of component mixture, it is also possible to estimate almost all continuous probability density functions. Gaussian mixture density is defined as

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \tag{4.39}$$

where x is a d-dimensional random variable, $N(x|\mu_k, \Sigma_k)$ is a multivariate normal distribution with mean μ_k and covariance matrix Σ_k and π_k are the so-called mixing coefficients for the k components of the distribution $p(x)$ which have to satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$ to form a convex combination of the mixture components Bishop [2012]. An illustrative example is shown in Fig. 4.7. We apply this clustering technique to our 1-dimensional temperature array and the output is the graph with the clusters.

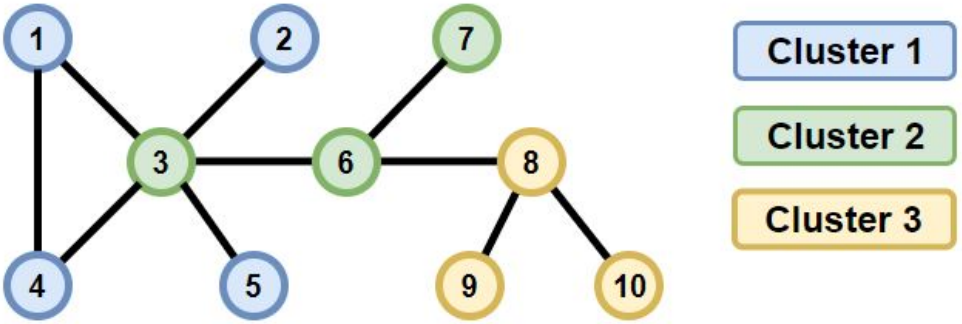

FIG. 4.7: Illustrative example of the application of the GMMs cluster technique to the graph with the temperatures of the IoT network.

4.4.3 Multiplex module and virtual network module

Multi-layer or multiplex networks can be defined as those that incorporate different connectivity channels, and describe systems that are interconnected with different categories of connection: each channel is represented by a sub-network or layer and

the same node can have different types of interactions and a different neighborhood in each layer. Let's formalize the concept of multiplex networks or multilayer networks.

Multi-layer network is a couple $M = (G, E)$ where $G = \{G_\alpha, \alpha = 1, \ldots, M\}$ is a family of graphs $G_\alpha = (N_\alpha, L_\alpha)$ (i.e., layers of the multi-layer network) and $E = \{E_{\alpha\beta} \subset N_\alpha x N_\beta, \quad \alpha \neq \beta\}$ is the set of interconnections between the nodes of the different layers. The E elements are called cross-layers, the L_α elements are called interlayer connections. The set of nodes of each layer is denoted by $N_\alpha = \{n_1^\alpha, \cdots, n_{N^\alpha}^\alpha\}$ and the *Laplacian matrix* of each layer is denoted by $A^{[\alpha]}$.

Once the multiplex is constructed, the connection between each graph and its layer is verified. In case they are not connected, as many virtual nodes are created as necessary to link the graph with the layer i*th*. The virtual nodes are vertices of the graph of other layers, however we act as if it were in that layer for practical purposes . In this way, the graphs will be connected and the control algorithms can be applied subsequently. If there is only one element in a layer, as many vertices of other layers will be virtualized as the amount of edges of a node in the initial graph. In the *Laplacian matrix* we will represent a virtualized node as 1^*.

Applying this definition now to the graph that we have constructed in the previous point, we are going to create a multiplex. Each one of the layers of the multiplex is going to be each one of the nodes of the graph that are in the same cluster. In order to assist in creating the multiplex we are going to create each of the adjacency matrices for each of the layers ($A^{[\alpha_1]}, A^{[\alpha_2]}$ and $A^{[\alpha_3]}$) as follows:

Layer 1 of the multiplex contains only the IoT nodes 1, 2, 4 and 5. Also, we have to create a virtual node to virtualize the IoT node 3.

$$A^{[\alpha_1]} = \left(\begin{array}{ccccc|ccccc} t_1 & 0 & 1^* & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_2 & 1^* & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1^* & 1^* & t_3^* & 1^* & 1^* & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1^* & t_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1^* & 0 & t_5 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \tag{4.40}$$

Layer 2 of the multiplex contains only the IoT nodes 3, 6 and 7.

$$A^{[\alpha_2]} = \left(\begin{array}{cc|ccccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & t_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & t_6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & t_7 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \tag{4.41}$$

Layer 3 of the multiplex contains only the IoT nodes 8, 9 and 10.

$$A^{[\alpha_3]} = \left(\begin{array}{ccccccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_8 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_9 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_{10} \end{array}\right) \tag{4.42}$$

In the *Laplacian matrix*, each layer of the multiplex has been marked with lines. This way, the next step in the algorithm is to build each layer of the multiplex. Fig. 4.8 shows a multiplex which illustrates the above techniques.

4.4.4 Projected data quality algorithm implementation

Once the control algorithms have been applied to the homogeneous multiplex data, the results are projected on to the layer $\alpha = 0$ (i.e., initial layer). In this way, we have a system similar to a Multi-input Multi-output (MIMO), but in its improved version because we implement consensus techniques in the cooperative algorithm. Thanks to the use of those techniques, the cooperative algorithm will use the clusters and neighboring sensors belonging to those clusters to compare the data and thus be able to self-correct any errors and find the wrong IoT nodes. Once the data is separated by homogeneous zones, the algorithm designed by Casado *et al.* [Casado-Vara et al.,

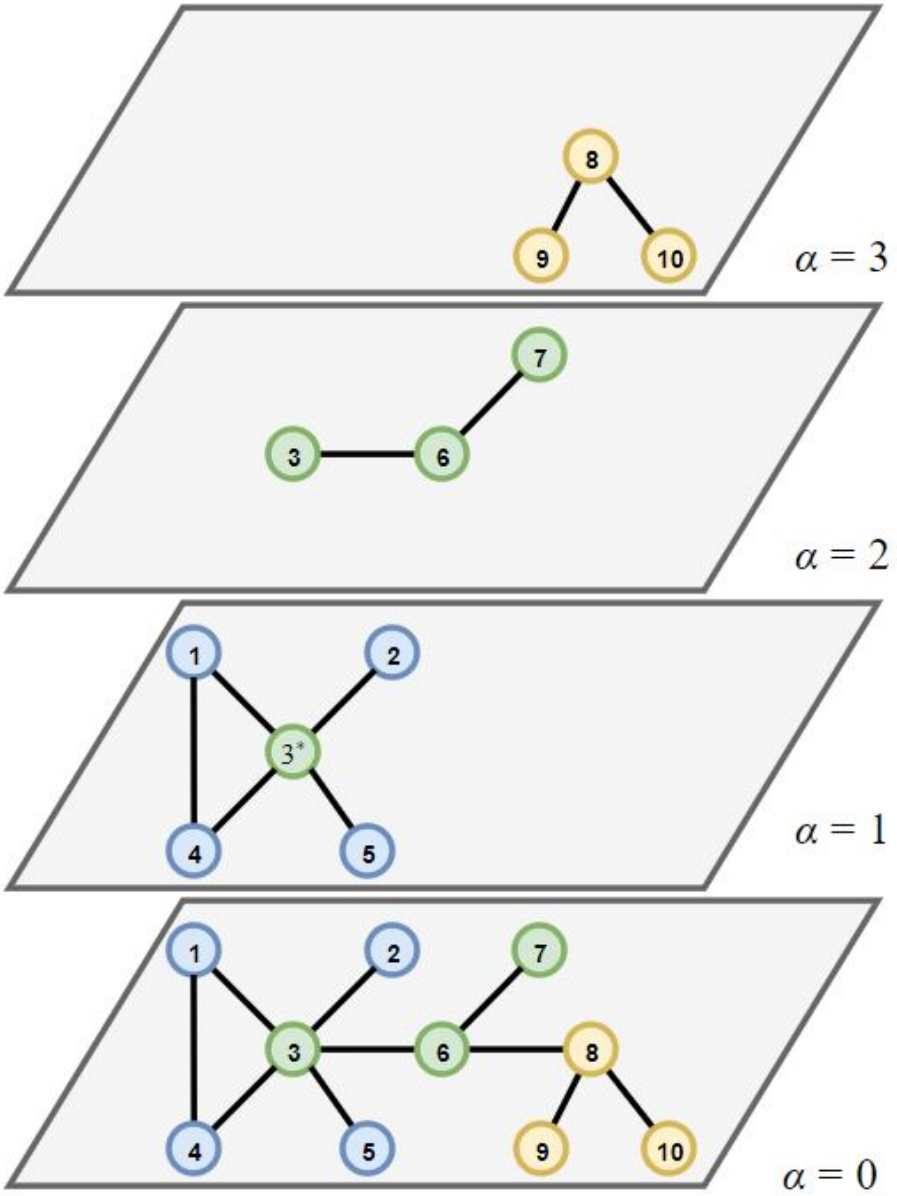

FIG. 4.8: Multiplex with 3 layers. In each of the layers are the IoT nodes that have similar temperatures according to the clustering algorithm used.

2018b] can be applied. A brief summary of this algorithm is presented in the next paragraphs. A detailed description of this algorithm is presented in 4.2.

This algorithm compares the neighborhood temperature of the sensors using a cooperative game based on game theory to detect errors in data and increase data quality gathered by the IoT nodes. In this algorithm, we want the neighbourhood coalitions to democratically decide the temperature of the main sensor. To do this, they will form coalitions that will decide on the final temperature of the IoT node, which will

be determined by whether they can vote or not in the process. From the characteristic function, if the value is 1(0), the coalition can vote (not vote) respectively. s_i is the main sensor with its associated temperature t_{s_i}, the characteristic function is built in the following way:

1. First, the average temperature of all the sensors is calculated:

$$T_{s_i}^k = \frac{1}{V}\sum_{i}^{V} t_{s_i} \tag{4.43}$$

   here $T_{s_i}^1$ represents the average temperature of the sensors' neighbourhood s_i (including it) in the first iteration of the game and V is the number of neighbours in the coalition.

2. The next step is to compute an absolute value for the temperature difference between the temperatures of each sensor and the average temperature:

$$\overline{T}_{s_i}^k = \left(\frac{1}{V}\sum_{i}^{V} \mid t_{s_i} - T_{s_i}^k \mid^2\right)^{\frac{1}{2}} \tag{4.44}$$

3. Using the differences in temperature values and the average temperature $\overline{T}_{s_i}^k$ (see eq.(4.44)) a confidence interval is created and defined as follows:

$$I_{s_i}^k = \left(T_{s_i}^k \pm t_{(V-1,\frac{\alpha}{2})}\frac{\overline{T}_{s_i}^k}{\sqrt{V}}\right) \tag{4.45}$$

   in Eq.(4.45) we use the Student's-t distribution with an error of 1%.

4. In this step we use a hypothesis test. If the temperature of the sensor lies within the interval $I_{s_i}^k$, it belongs to the voting coalition, otherwise, it is not in the voting coalition:

$$u^k(s_1,\ldots,s_n) = \begin{cases} 1 & \text{if } t_{s_i} \in I_{s_i}^k \\ 0 & \text{if } t_{s_i} \notin I_{s_i}^k \end{cases} \tag{4.46}$$

5. The characteristic function will repeat this process iteratively (k is the number of iterations) until all the sensors in that iteration belong to the voting coalition. In each iteration k, the following payoff vector of the coalition S_j (with $1 \leq j \leq n$ where n is the number of sensors in the coalition) is available in step k ($PV(S_j^k)$):

$$PV(S_j^k) = (u^k(s_1), \ldots, u^k(s_n)) \text{ where } \sum_i^n u^k(s_i) \leq n \tag{4.47}$$

The stop condition of the game iterations is $PV(S_j^k) = PV(S_j^{k+1})$ the process end. That is, let $PV(S_j^k) = (u^k(s_1), \ldots, u^k(s_n))$ and let $PV(S_j^{k+1}) = (u^{k+1}(s_1), \ldots, u^{k+1}(s_n))$. The iteration process ends when both payoff vectors contain the same elements. This process is represented by the following equation:

$$\begin{cases} u^k(s_1) = u^{k+1}(s_1) \\ \vdots \\ u^k(s_n) = u^{k+1}(s_n) \end{cases} \tag{4.48}$$

4.5 Improving robustness in IoT networks

This section details an algorithm to increase the fault tolerance of the IoT network. In this way, the smart buildings monitored and controlled by these robust networks are more efficient, therefore, they are energy-saving buildings.

4.5.1 Fault-tolerant adaptive control for IoT networks with external disturbances and uncertainties

This section shows the adaptive control algorithm developed in this article. The main purpose of this algorithm is to control the temperature of the smart building. For this purpose, the control algorithm relies on 3 mechanisms to optimize temperature control: 1) Cooperative control algorithm. This mechanisms improves data quality gathered by the IoT nodes and finds false data. 2) State prediction. This mechanisms receives the error the IoT nodes are making with respect to the desired temperature and predicts the error they will make in the future. 3) Controller. In this control algorithm a PID controller is used to control the temperature. But this PID input is being optimized by the 1) and 2) mechanisms. This algorithm is shown in Fig. 4.9.

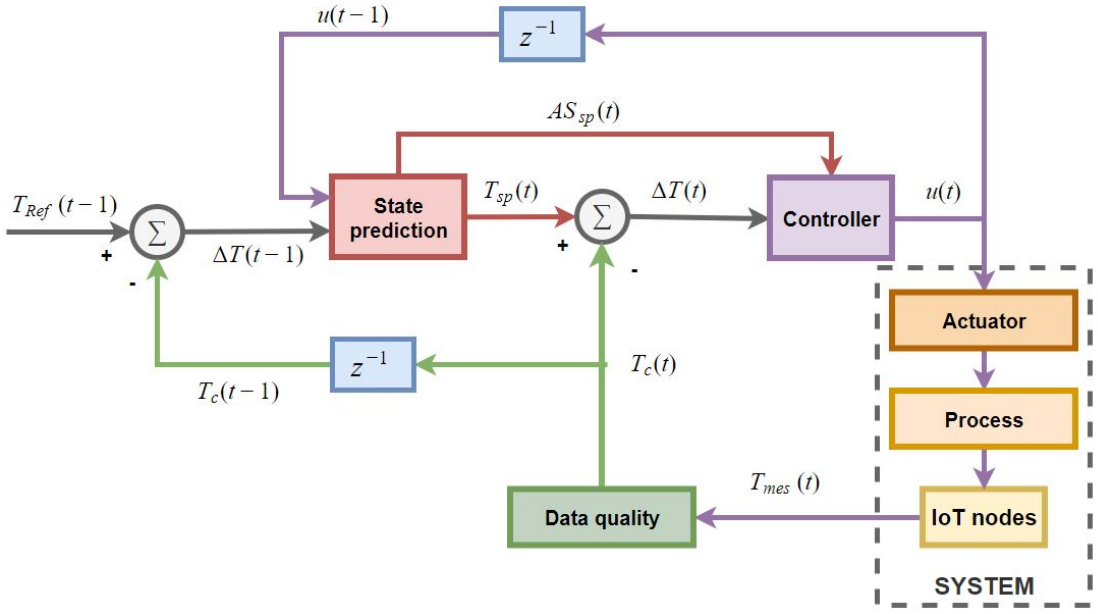

FIG. 4.9: This algorithm forecasts the Iot nodes accuracy state through the adaptive control algorithm at the time-interval.

4.5.2 Reference input

The Reference input (RI) in our algorithm is the desired temperature value. In this algorithm the RI is located in a control room where the building manager will set the temperature that the building should have. Then the algorithm controls the actuators (thermostats) so the building reaches that temperature and remains stable.

4.5.3 State predictor

This step of this algorithm is designed to predict that IoT nodes will be in faulty state in time t and then output the predicted temperature ($T_{sp}(t)$). In this subsection we describe in detail the operation of the state predictor.

4.5.3.1 Initial accuracy state

There is a need to design a range of degradation of precision given in percents. This is according to data collected by the algorithm we had carried out in previous research Casado-Vara et al. [2019a]. This range are the random variable X_n which determines the current accuracy states of the system in relation to IoT nodes error. Therefore, the IoT nodes' current states are $X_n = \{A = high\ accuracy,\ B = accurate,\ C = low\ accuracy,\ F = failure\}$. Below, table 4.2 has the selected for every parameter.

X_n	IoT node accuracy state	Error (%)
A	High accuracy	$e \leq 10$
B	Accurate	$10 < e \leq 20$
C	Low accuracy	$20 < e \leq 35$
F	Failure	$e \geq 35$

TAB. 4.2: IoT nodes accuracy state.

4.5.3.2 Prediction step

The controller's first action is the predictive step. This algorithm step uses the transition matrix of the proposed model. Let $z^{(t)} : T^{g(t)} \longrightarrow z^{(t)}(T^{g(t)}) = T_{sp}(t)$ be the forecasting feature of precision states (i.e., Prediction step) for every period of time t and let $t + k$ where $k \in \{1, 2, \cdots\}$ be the predicted time. Given $t^{\delta}_{i,j} \in T^{\delta}$, the function u of the controller is described next:

$$z^{(t+k)}_{ij}(t^{g}_{i,j}) = max\{\mathbb{P}_{t^{g(t+k)}_{i,j}A}, \mathbb{P}_{t^{g(t+k)}_{i,j}B}, \mathbb{P}_{t^{g(t+k)}_{i,j}C}, \mathbb{P}_{t^{g(t+k)}_{i,j}F}\} \tag{4.49}$$

Let $z^{(t)}(T^{g}) = T_{sp}(t)$ be the state of precision matrix given by the predictive function. The result of this function is the precision state of the IoT nodes at the moment t.

4.5.3.3 Temperature of the prediction step

In the state predictive step the accuracy state that the sensor is going to have is predicted from the measurement error of the sensor with respect to the reference input ($\Delta T(t-1)$). Using the accuracy error of the sensor, an adjustment factor is used to predict the temperature the sensor will have in time t. Then the state prediction step temperature is calculated as follows:

$$T_{sp}(t) = \begin{pmatrix} 1.05 & 0 & 0 & 0 \\ 0 & 1.15 & 0 & 0 \\ 0 & 0 & 1.25 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot T_c(t-1) \tag{4.50}$$

4.5.4 Data quality

This algorithm compares the neighborhood temperature of the sensors using a cooperative game based on game theory to detect wrong data and increase the quality of data gathered by the sensors. In this algorithm, we expect neighborhood associations to democratically determine the main sensor temperature. To this end, coalitions are

formed to determine the final temperature of the IoT node, which is calculated according to whether or not they can vote in the process. Since the characteristic function, when the number is 1(0), each candidate can cast a vote (not vote). s_i is the head IoT node with its temperature related t_{s_i}, the characteristic function is created like this:

1. The mean temperature of all IoT devices is determined in the first place:

$$T_{s_i}^k = \frac{1}{V} \sum_{i}^{V} t_{s_i} \tag{4.51}$$

   here $T_{s_i}^1$ represents the average temperature of the sensors' neighbourhood s_i (including it) in the first iteration of the game and V is the number of neighbours in the coalition.

2. The next step is to compute an absolute value for the temperature difference between the temperatures of each sensor and the average temperature:

$$\overline{T}_{s_i}^k = \left(\frac{1}{V} \sum_{i}^{V} \mid t_{s_i} - T_{s_i}^k \mid^2 \right)^{\frac{1}{2}} \tag{4.52}$$

3. Use the variations in the temperature against the mean temperature $\overline{T}_{s_i}^k$ (see eq.(4.52)) a confidence interval is defined in the following way:

$$I_{s_i}^k = \left(T_{s_i}^k \pm t_{(V-1,\frac{\alpha}{2})} \frac{\overline{T}_{s_i}^k}{\sqrt{V}} \right) \tag{4.53}$$

   in Eq.(4.53) we use the Student's-t distribution with an error of 1%.

4. In this step we use a hypothesis test. If the temperature of the sensor lies in the interval $I_{s_i}^k$, it belongs to the voting coalition, otherwise, it is not in the voting coalition:

$$u^k(s_1, \ldots, s_n) = \begin{cases} 1 & \text{if } t_{s_i} \in I_{s_i}^k \\ 0 & \text{if } t_{s_i} \notin I_{s_i}^k \end{cases} \tag{4.54}$$

5. The characteristic function is repeating this procedure iteratively until all IoT nodes on that iteration are part of the voting alliance. In each k iteration the following payoff vector of the coalition S_j (with $1 \leq j \leq n$ where n stands for the number of IoT nodes in the combination) at the step k ($PV(S_j^k)$) is available:

$$PV(S_j^k) = (u^k(s_1), \ldots, u^k(s_n)) \text{ where } \sum_i^n u^k(s_i) \leq n \tag{4.55}$$

The stop condition of the game iterations is $PV(S_j^k) = PV(S_j^{k+1})$ the process end. That is, let $PV(S_j^k) = (u^k(s_1), \ldots, u^k(s_n))$ and let $PV(S_j^{k+1}) = (u^{k+1}(s_1), \ldots, u^{k+1}(s_n))$. The iteration process ends when both payoff vectors contain the same elements. This process is shown in the following equation:

$$\begin{cases} u^k(s_1) = u^{k+1}(s_1) \\ \vdots \\ u^k(s_n) = u^{k+1}(s_n) \end{cases} \tag{4.56}$$

So the game is able to define its solution in the next subsection.

This algorithm is described in Casado-Vara *et al.* Casado-Vara et al. [2019a].

4.5.5 Controller

A PID controller is the most common device/algorithm in industrial control loops applications Åström et al. [2006] Sung et al. [2009].

Basically, obtains de control signal apply to a system from the error between the desired set point (SP) and the measured process value (PV) Åström et al. [2006].

The main aim of the PID controller is to minimize the error (SP-PV) thanks the control inputs tuning. The standard PID topology and, usually their different developments, involves three actions or terms of control: the proportional, the derivative and the integral actions Michael and Mohammad [2005].

4.5.6 PID controller format

As previously mentioned, there are a lot of PID controller topologies, but during this work the standard format presented in equation 4.57 has been employed.

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right] \tag{4.57}$$

where $'u'$ is the signal control and $'e'$ is the error value given from $'e = SP - y'$ (difference between the setpoint $'SP'$ and process value conditioned $'y'$). The other terms are the controller parameters: proportional $'K'$, integral $'Ti'$ and derivative $'Td'$.

4.5.7 PID controller tuning in closed-loop

This section show the PID controller tuning method used in this work based on close-loop tuning. Firstly, it is shown the general method to obtain the controller parameters, and then, the used procedure to achieve the response characteristics. To end, the common expressions employed to calculate the PID controller parameters are presented.

4.5.7.1 General procedure to calculate parameters

Two are the steps to obtain the controller parameters to achieve its tuning with close loop method:

- The first step is to achieve a system response under a permanent oscillation state. Then, some characteristics of its response must be measured.
- With the previous achieved response, the right expressions must be used to obtain the controller parameters to obtain a plant response with a desired specifications.

4.5.7.2 Response characteristics obtaining in closed-loop PID tuning

There are a lot of different methods to obtain the tuning controller parameters. The relay-feedback methodology has been used in this work, described by Aström and Hägglud in Åström et al. [2006].

This method achieve results similar to the ones obtained with the traditional one developed by Ziegler–Nichols Visioli [2006]. However it has two very important advantages: the system is far from the unstable region, and the tuning could be acomplished at any moment and for any work operation point.

4.5.7.3 Obtaining controller parameters

The controller tuning parameters are calculated with the Tc and Kc specifications measured during the previous step. There are a lot of expressions, that have been developed for many researchers during the years Åström et al. [2006], Visioli [2006],

Machón González et al. [2011]. The main objective is to obtain the rigth values of items like: overshoot, peak response, settling time, rise time and so on. Then, it is possible to improve the system specifications, and achieve a more robust control under a specific criteria (load disturbance or set point). For this research, the Ziegler–Nichols method was used. This method is defined for Load Disturbance rejection criteria with the aim to reduce the perturbations effect.

4.6 Conclusions

Algorithms presented in this chapter allow to optimize energy consumption in smart building monitored and controlled with IoT networks. These reductions in energy consumption can be achieved by, for example, the increase in data quality and the prediction of future states of accuracy increase the monitoring of IoT networks. On the other hand, the IoT slicing technique optimizes the control of the IoT network by converting heterogeneous data into homogeneous data. This results in a significant increase in the effectiveness of the proposed control algorithms.

These algorithms are designed in such a way that the system adapts to the technical characteristics of the smart building, without the need of any configuration whatsoever, simply by developing some sensors to obtain data from the environment.

These algorithms are able to:

- They improve the monitoring and control of IoT networks.
- The system can even obtain temperature data using IoT devices in the smart building, this means that it is not necessary to deploy new devices.
- These algorithms adapts dynamically to all kind of IoT networks in smart buildings. This is due to the functions of these algorithms such as increasing data quality, predicting future states of accuracy, transforming heterogeneous data into homogeneous data and increasing tolerance to network failures.

On a technical level, the main advantage of these algorithms lies in its modular structure. This facilitates the incorporation of artificial intelligence (AI) techniques to improve the operations of these algorithms. Notice that AI techniques contribute to more efficient control algorithms that allow to improve the energy saving in smart buildings.

Chapter 5

Case studies

VNiVERSiDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Chapter 5

Case studies

5.1 Introduction

This chapter describes the case studies (real or simulations) whose objective has been to evaluate the performance of the architecture proposed in this thesis and the algorithms designed to improve the monitoring and control of the dynamic networks of IoT. Each case study is designed to test each of the aspects and functionalists of the architectures and algorithms. These techniques and technologies have as their main objective to improve energy saving.

The aim of the case studies is to evaluate each of the states of the research conducted. The effectiveness of the architecture and algorithms designed in this work has been tested first. The optimization has been carried out in simulations and in case studies in smart buildings. These case studies are designed to test whether the monitoring and control of IoT networks is improved in different ways. All these case studies are designed following the proposals of the chapter 3 and chapter 4.

The first case study has been designed to test the effectiveness of the proposed architecture by focusing on the routing algorithm designed for this platform. In this case study, a simulation has been designed to demonstrate the effectiveness of the designed algorithm.

The second case study is designed to test the efficiency of the data quality algorithm. In this case study we present a real case study with temperatures collected by an IoT network in the R&D building of the University of Salamanca to test the efficiency of the algorithm.

The third case, a series of simulations was designed with the objective of demonstrating the effectiveness of the predictive algorithm of future states of precision. In this case study we present a real case study with temperatures collected by an IoT network in the R&D building of the University of Salamanca.

The fourth case study was tested in a smart building with heterogeneous data collected by an IoT network, and has been designed with the aim of testing the efficiency of the new technique developed to transform heterogeneous data into homogeneous data.

The fifth case study was designed in a smart building with the aim of saving energy and testing the efficiency of the algorithm that increases the fault tolerance of the IoT network.

These case studies in real environments have been carried out at the R&D smart building and in a supermarket both in Salamanca. These smart buildings have been monitored with a IoT network to collect temperature data to design experiments and test the effectiveness of the architecture and algorithms designed in this work. In the case of studies based on simulations, these simulations have been designed to test all aspects of the algorithms and demonstrate their efficiency.

The rest of the paper is organized as follows: Data routing in IoT architectures case study is shown in section 5.2. Case study II presents the data quality algorithm in section 5.3. Section 5.4 present the framework to test the future accuracy states prediction case study. IoT network sliding case study is shown in section 5.5 and case study V test the effectiveness of the fault-tolerant algorithm in section 5.6. Finally, conclusions are shown in section 5.7.

5.2 Case study I: Data routing in IoT architectures

5.2.1 Introduction

A Networked Control System (NCS) is a control system wherein the control loops are closed through a communication network [Zhang et al., 2016]. Their advantages include low installation and maintenance costs, flexibility and reduced wiring [Ge et al., 2017]. These benefits make NCSs applicable to a wide range of fields [Qiu et al., 2016]. However, NCS's weaknesses are caused by delays in random communications and packet loss. In terms of current industrial applicability, traditional point-to-point centralized control is unsuitable since it does not meet the new requirements such as modularity, decentralized/distributed control, quick and easy maintenance and low cost. For these reasons, in recent years NCS has been a major focus of attention in both academic research and industrial applications, contributing to significant progress in this field [Hespanha et al., 2007]. IoT networks usually consist of many sensors, as well as controller and actuator nodes. These distributed IoT nodes compete to send their data to the network. There is a need to implement a control system that monitors and

controls the sending of packets from the IoT nodes to the network. In addition, IoT networks generate large amounts of data continuously. The volume of data makes it very difficult to monitor and control IoT networks. To control the IoT network it is necessary to search within the databases. Causing a delay in the functioning of the control system within the IoT network. This also entails high levels of consumption of energy and resources.

Although the problem of the limitation of communications in networks has been well studied, there are relatively few works on the optimization of queuing analysis. Furthermore, none of those works incorporate properties of the network into the control system [Sun et al., 2018] [Liu et al., 2018] [Casado-Vara et al., 2018c]. The problem remains unresolved; the properties of IoT networks continue to affect system performance to a large degree. Thus, these properties must be taken into consideration by the control algorithm used by the networks [Tan et al., 2015]. Most of the research is based on optimizing energy consumption for improved performance of NCSs. However, the use of NCSs with IoT network properties, including queue analysis, service rate and packet dropout, must be subjected to further research. In order to make the monitoring and control of the IoT network more effective, some researchers proposed new techniques that increase the speed at which big data databases are searched. Zhou *et al.* designed a framework to bridge multi-target query needs between users and the data platform, including required query accuracy, timeliness, and query privacy constraints [Zhou et al., 2018]. Another research proposed the use of binary hashing for greater search speed; Cao *et al.* reviewed and compared those hash techniques through different experiments [Cao et al., 2018].

In this thesis proposes a new IoT architecture that covers the research gaps in the monitoring and control of the queues that control the sending of packets from the IoT nodes to the big data databases. To optimize the process of sending data over the network, a novel adaptive control algorithm is proposed in this work. The adaptive control algorithm analyzes the queues within the network nodes and the server, improving their performance and impeding the queue from becoming saturated. If the queue is saturated, the adaptive control algorithm will make the necessary changes to the network properties in order to desaturate the queues. In addition, we proposed a novel hashmap-based search system to speed up the search time in big data databases. However, one of the main problems of IoT networks is security. To secure communications, we have integrated blockchain technology in our architecture. Although this is not the main objective of our research, blockchain technology stores the hashmap that is used for speed searches in the big data databases. This improves the monitoring and control efficiency of the IoT network.

The objective of this case study is to test the effectiveness of the architecture by focusing on the routing algorithm. In addition, we want to test if the hashmap search improves the response speed of IoT network monitoring. The simulations designed for this case study have been carefully chosen to test the effectiveness of the architecture, the routing algorithm and the speed up of hashmap-based searches. In order to test the effectiveness of the routing algorithm, a case study has been designed in which the number of blocks sent to the network is randomly increased or decreased. In this way, the efficiency of the algorithm is tested in a case study with randomness (i.e., trying to simulate a real case). The effectiveness of the algorithm will be tested when the tail collapses, which is the worst case. On the other hand, has been designed the case study of the search in hashmap in such a way that in each of the tests to prove its effectiveness, we will work with hashmap bigger in each test.

5.2.2 Simulation setup and description

In order to verify the efficiency of the proposed smart routing algorithm in the optimization of energy consumption, a monitoring and evaluation experiment has been designed. In this experiment variables have been designed and they will be used to evaluate the effect of this algorithm produces in the routing algorithm of the proposed architecture. This approach consists in monitoring the package sending in the smart building new architecture and its new routing algorithm in order to validate the results obtained.

The experiment was divided into two stages (routing algorithm and speed up searches) and this experiment allows to evaluate the efficiency of these two methods. The choice of data in which the experiment was performed was due to worst cases for our new algorithm to prove its efficiency. Both simulations were monitored to collect data and prove the operation of our new routing algorithm and our new speed up searches to validate our methods.

The simulation detailed in this subsection shows the functioning of the network with the queuing model and the adaptive control algorithm proposed in this thesis. The model is an open network of self-adaptive Jackson, in which each of the queuing block queues is a $M|M|1$ queue that sends its block to the block queuing system.

Let's assume the value of some of the parameters that we will use in this example:

- Let's assume that, on average, the value of $C_k = 30\ C_h$. The value of these parameters was calculated from the mean value of the experimental data.
- The time unit t is 1 hour.

- Number of block queuing nodes = 4 and one block queuing system.
- $\gamma_i \in$ random value between $\{1, \cdots, 10\}$ where $i \in$ {set of the block queuing nodes}.

On the other hand, the speed up searches experiment was developed comparing two arrays, one of the with random data into a current data base and the another one with random data in a hashmap. The lenght of these arrays goes from 2^{15} to 2^{20}. The main objetive of this experiment is to prove this new searching method for a random value in both arrays of data.

5.2.3 Conclusion of the integration of the proposed architecture and the new routing algorithm for the case study I

This case study is directly oriented to the main use of the architecture and the smart routing algorithm and is focusing on the field of energy optimization. Advances in this area can have a major economic impact on smart building and the environment. Therefore, this case study seeks to have an impact on the environment and on the citizen, reducing energy consumption by the optimization of the routing technique of the current IoT architectures.

Thanks to the use of the proposed architecture and our new routing algorithm, the development of a system to model a case study of energy optimization in the smart buildings have been greatly simplified. On the one hand, the use of the architecture allows the deployment of the IoT devices to collect data. On the other hand, this new routing algorithm and the speed up monitoring system allow us to optimize the energy saving in smart building monitoring by IoT networks.

5.3 Case study II: Data quality based on consensus

5.3.1 Introduction

Dynamic IoT Networks have become important in the last years and nowadays are present in practically all the sectors of our society [Haibo and Fang, 2015]. Their great capacity to gather data may facilitate the construction of smart environments, allowing for a flexible analysis of processes that occur in the environment and the services offered to users. There are many advances in IoT architecture however, the efficient management of the data generated by them is still a challenging aspect. Data management is complicated because the data may sometimes be inconsistent due to different reasons (i.e. it is difficult to determine if data is reliable or if sensors are accurate, etc.). Therefore, there is a growing need for new IoT architectures which would merge data from heterogeneous sensors, and smart management of the collected data.

The state of the art contains some architectures that allow to merge data from IoT [Gungor et al., 2009, Patel and Pandey, 2010]. There are also some novel frameworks that define methods for integrating dynamic and self-adaptable heterogeneous IoT [Tapia et al., 2010b], and manage data obtained from IoT [Rodríguez et al., 2015]. Other architecture proposals have demonstrated that the accuracy of IoT can be improved with the use of artificial neural networks [de Paz et al., 2013]. Another work presents a multi-agent system that automatically processes and merges information in heterogeneous distributed IoT [Bajo et al., 2015]. However, some frameworks are designed for very specific purposes and their functionality is limited [Alonso et al., 2013, Tapia et al., 2010a, 2009]. It is also difficult to merge and manage the data obtained from heterogeneous IoT [Bowman and Steinberg, 2008].

In this thesis, we present an algorithm that will ensure the robustness and reliability of the data collected by IoT. In our approach we apply game theory (GT) to data obtained from a IoT. The related work on IoT and GT, shows a research gap in this area. To the author's knowledge no other study proposed the application of game theory to the problem of data quality and false data detection. Thus, in this thesis we propose a game to solve the problem of data reliability. Our game is distributed and self-organized so that it can work in a IoT regardless of the number of sensors, the architecture of the IoT or the type of sensors to which the game is applied.

The design of the game fulfills the following needs: it is capable of recognizing the neighbourhood in which it is implemented (the environment of the sensor, which is defined further on in this thesis). The game also identifies the possible coalitions that

can be formed between the neighbours. Finally, the temperature is determined by the winning coalition for the sensor to which the game has been applied. Moreover, the convergence of the game is monitored in order to find the Nash equilibrium, which is defined for the data quality algorithm.

For these reasons, this case study carried out in the R+D+i smart building of Salamanca is designed to test the efficiency of the data quality algorithm. The data needed to perform this experiment has been collected by an IoT network in the smart building. For this case study the temperature has been used as a parameter, but since the algorithms have been designed with a high degree of abstraction, any type of environmental variable could be used, and the algorithms would continue to be effective by increasing the quality of the data and detecting malfunctioning IoT nodes. This means that the data that will be integrated into the architecture and for the use of other algorithms and techniques have a better quality and there are no false data. Therefore, the efficiency of the algorithms is superior.

5.3.2 Experimental setup

In order to verify the efficiency of this data quality algorithm to improve data and detect false data (i.e., find inaccurate IoT devices). The experiment was conducted in the 2^{nd} R+D+i smart building of the University of Salamanca which is shown in Fig. 5.1. By conducting the case study in this flood with the same technical characteristic, the proposed data quality algorithm can be validated in a one-day time period. We don't consider outdoor temperature. This experiment took place in January 15, 2018 after several simulations to test its efficiency.

5.3.3 General description of the experiment

To test the proposed model we have chosen a building. At the time the sensors measured the temperature, the thermostat of the building showed 22°C. A mesh was used to place the sensors in the surface (Figure 5.1). With the help of laser levels, the sensors were placed vertically one in every different room. A total of 25 IoT nodes were deployed.

The type of the sensor that was deployed was a combination of the ESP8266 microcontroller in its commercial version "ESP-01" and a DHT11 temperature and humidity sensor. The sum of both allows us for greater flexibility when collecting data and adaptability to the case study, since the DHT11 sensor is designed for indoor spaces (it has an operating range of 0°C to 50°C) according to its datasheet [`http://www.micropik.com/PDF/dht11.pdf`]. The microcontroller collects data from this sensor

FIG. 5.1: Map of the building with all the sensor placed, a sample of an indoor surface for testing our model and a sample of a sensor.

through the onewire protocol and communicates it to the environment via Wi-Fi using HTTP standards and GET/POST requests. The ESP-IDF programming environment provided by the manufacturer of the microcontroller, was used to programme the device.

The sensors had been collecting data at 15 minute intervals, for an entire day. For the analysis we selected the data collected by the sensors in the following time interval $2018-01-15T09:00:00Z$ and ended on $2018-01-15T16:00:00Z$. A specific moment has been selected since the game that has been defined is static and not dynamic (i.e., it does not process the data in a temporal evolution). Below, a statistical summary of the measurements that were made with the sensors is presented in Table 5.6.

TAB. 5.1: Statistical table of measurements of the IoT nodes case study II.

Timestamp start	**Total timestamp**	**Min temp**	**Max temp**	**Mean**	**Standard deviation**
2018-01-15T09:00	07:00:00Z	20 °C	22.6°C	21.39°C	0.41°C

5.3.4 Conclusion of the integration of the proposed data quality algorithm for the case study II

The integration of the data quality algorithm has allowed the system to improve the quality of the data and find the malfunctioning IoT nodes. In addition, the use of this

algorithm that obtain data from the IoT network has made it possible to design a case study in which to test the efficiency of the algorithm by adding disturbances to the experiment, such as raising the heating or opening windows. The main objective of this case study is to test whether the algorithm is adaptive and self-corrects the false data it finds.

The efficiency of this algorithm is fundamental for the rest of the algorithms, since the following case studies designed to test the effectiveness of the rest of the algorithms are based on the fact that the data quality algorithm is effective. Therefore, this case study has been designed and performed with a high degree of attention to detail.

5.4 Case study III: Future states prediction

5.4.1 Introduction

With the development of communications techniques, network topologies and control methods, networked control systems have received increasing attention in the past decades due to its widespread applications [Hespanha et al., 2007]. Meanwhile, because an Internet of Things network is usually shared by multiple sensor, controller and actuator nodes, these IoT nodes collect data from a wide variety of buildings. There is a need for IoT network monitor and control to improve the detection of sensors that are collecting false data or malfunctioning [Mo et al., 2010]. This thesis presents a new predictive feedback control algorithm for handling the predictive management of a huge amount of IoT nodes. There is a need to implement a control system that monitors and controls the accuracy states of the IoT nodes. In this way it is possible to ensure confidence in the data collected by the IoT network. Discrete-time control mainly studies the performance of the system in a discrete-time interval rather than a continuous time interval. The discrete-time control problems for linear systems have been investigated such as linear systems [Amato et al., 2006] [Amato et al., 2010] [Polyakov et al., 2016]. Meanwhile, the studies on the discrete-time control of nonlinear system have also been carried out for triangular systems [Korobov et al., 2013], nonlinear dynamical networks [Hui et al., 2008], etc. Discrete-time control techniques have been applied for many practical applications, for instance, multiagents systems [Khoo et al., 2014] and secure communications [Perruquetti et al., 2008]. Feedback nonlinear systems representing a class of nonlinear control systems have been widely concerned [Li et al., 2015] [Li and Yang, 2018]. The problem we want to deal with in this work is the predictive maintenance of IoT networks in continuous-time. In this way it is possible to improve the reliability in the monitoring and control of IoT networks as it is done in continuous-time. By using continuous-time Markov chains to predict the future states of sensor accuracy, IoT networks can improve the data quality since they will always be working in the best possible condition.

Motivated by the above observation, this thesis proposes a new feedback control algorithm to improve predictive maintenance of the IoT networks. The algorithm finds the IoT nodes that are collecting false data or malfunctioning. To optimize the process of monitoring and control of the IoT network, a novel application of the continuous-time Markov chains is used. We predict the accuracy future states of the IoT nodes and in case that prediction is that the sensor will be in fault state, after the time control period has expired, the controller sends the signal that this IoT node has to be replaced. Moreover, if an IoT node has to be replaced, the control algorithm creates a virtual sensor in

that position. This virtual sensor estimates the temperature of that sensor based on the temperature of its neighboring nodes. In this way, the IoT network collects data in continuous-time range without any loss of reliability in the data due to malfunction of the IoT devices.

Although the problem of data quality and false data detection has been widely studied [Casado-Vara et al., 2019a] [Pipino et al., 2002] [Wang, 1998], the aforementioned works on data quality and false data detection are concerned with discrete-time systems, and the corresponding results for continuous-time systems are relatively few. In fact, continuous time control systems have already been applied in a wide range of areas, such as feedback control of nonlinear systems [Liu et al., 2018] [Zhang and Lin, 2012], time-delay communications [Zhang and Lin, 2015], control of marine surfaces [Zhang and Yang, 2018] and neural networks [Wang et al., 2015]. The control algorithms have the following challenges in the field of data quality and predictive maintenance of IoT networks.

1. For predictive maintenance in continuous time it is necessary to solve complex differential equations with initial conditions and boundaries that change in each iteration.

2. Algorithms to increase data quality and find false data can produce false positives. It is important to discriminate between a hot (cold) spot and a faulty sensor.

In this thesis, we deal with the research gaps in the monitoring and control of continuous-time networked systems with multiple IoT devices, aiming at presenting an improved control algorithm to find the maximum allowable efficiency in predictive maintenance. A unified continuous-time hybrid control system model is presented with an algorithm to improve data quality and false data detection and a feedback control algorithm to predict the accuracy state of the IoT devices. The output of the data quality algorithm is the input of the predictive feedback control algorithm.

This case study is going to test the efficiency of the precision state prediction algorithm. For this case study we have designed a real experiment prepared specifically to test all possible cases that may occur in the useful life of this algorithm. This case study is designed to test the control algorithm that uses the future state prediction algorithm. For the Markov chains to be used in this case study, the necessary coefficients have been calculated by averaging the coefficients of useful life and permanence in states of precision.

5.4.2 Experimental setup

Let's suppose for this simulation example that the sensors throughout their useful life can have 4 accuracy states (A=high accuracy, B=accurate, C=low accuracy, F=failure). The probability of a sensor that is in state A at instant t moves to state F in the time interval $(t, t+\Delta t)$ is $0.1\Delta t + o(\Delta t)$, if it is in state B it is $0.2\Delta t + o(\Delta t)$ and if it is in state C it is $0.5\Delta t + o(\Delta t)$. Let's assume for this simulation that time of the sensors are in state A is an exponential time of 2.1 in state A and 1.2 in state B.

From A in a time interval $(t, t+\Delta t)$ the sensor can pass to F with probability $0.1\Delta t + o(\Delta t)$. If ξ is the time the sensor stays at A, you have it:

$$P(\xi > t+\Delta t | \xi > t) = \frac{e^{-2.1(t+\Delta t)}}{e^{-2.1t}} = e^{-2.1\Delta t} = 1 - 2.1\Delta t + o(\Delta t) \tag{5.1}$$

Therefore, the eq. (5.7) is the probability of continuing in A at instant t_{i+1} if it was in A at instant t_i. Then, the probability of going to B between t and $t+\Delta t$ is

$$1 - ((1 - 2.1\Delta t + o(\Delta t)) - (0.1\Delta t + o(\Delta t))) = 2\Delta t + o(\Delta t) \tag{5.2}$$

This way, at successive stages, we reach the calculation that the transition matrix between t and $t+\Delta t$ are shown in table 6.1.

	A	**B**	**C**	**F**
A	$1 - 2.1\Delta t + o(\Delta t)$	$2\Delta t + o(\Delta t)$	$o(\Delta t)$	$0.1\Delta t + o(\Delta t)$
B	0	$1 - 1.2\Delta t + o(\Delta t)$	$\Delta t + o(\Delta t)$	$0.2\Delta t + o(\Delta t)$
C	0	0	$1 - 0.5\Delta t + o(\Delta t)$	$0.5\Delta t + o(\Delta t)$
F	0	0	0	1

TAB. 5.2: For this simulation, we have assumed that state F is absorbent. That is, for the sensor to move from F to any other state, it needs to be repaired by a maintenance worker.

Thus,

$$P'(0) = \begin{pmatrix} -2.1 & 2 & 0 & 0.1 \\ 0 & -1.2 & 1 & 0.2 \\ 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.3}$$

which may be expressed as follows:

$$P'(0) = \frac{1}{0.504}\begin{pmatrix}1 & 1 & 2 & 1\\ 1 & 0.8 & 0.9 & 0\\ 1 & 0.56 & 0 & 0\\ 1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0 & & & \\ & -0.5 & & \\ & & -1.2 & \\ & & & -2.1\end{pmatrix}\begin{pmatrix}0 & 0 & 0 & 0.504\\ 0 & 0 & 0.9 & -0.9\\ 0 & 0.56 & -0.8 & 0.24\\ 0.504 & -1.12 & 0.7 & -0.084\end{pmatrix} \tag{5.4}$$

Thus:

$$P(t) = \frac{1}{0.504}\begin{pmatrix}1 & 1 & 2 & 1\\ 1 & 0.8 & 0.9 & 0\\ 1 & 0.56 & 0 & 0\\ 1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}1 & & & \\ & e^{-0.5t} & & \\ & & e^{-1.2t} & \\ & & & e^{-2.1t}\end{pmatrix}\begin{pmatrix}0 & 0 & 0 & 0.504\\ 0 & 0 & 0.9 & -0.9\\ 0 & 0.56 & -0.8 & 0.24\\ 0.504 & -1.12 & 0.7 & -0.084\end{pmatrix} \tag{5.5}$$

For example, the term $p_{AF}(t)$ represents the probability that a sensor that starts its useful life at stage A, is in failure at time t, so:

$$P(\text{Life span} \leq t) = p_{AF} = 1 - \frac{0.9}{0.504}e^{-0.5t} + \frac{0.48}{0.504}e^{-1.2t} - \frac{0.084}{0.504}e^{-2.1t} \tag{5.6}$$

In Figure 5.2 the graphical representation of the Markov chain is presented. Probabilities of changes in the accuracy states of the sensors are shown in the Table 6.1. In the example simulation presented in this section it can be found that sensors from any of the precision states (i.e., A,B,C) can move to the fault state (F). While from state A it goes to state B, and from state B to state C. This is so, since in this example we assume that the sensor from any of its precision states can fail, while, we assume that a high accuracy sensor (A), has to go through the precise state (B) before moving to the low accuracy state (C).

Given the Markov chain used for this simulation with transition matrix given by the eq. (5.10), the stationary paths given by the probabilities of change of precision state of the sensors are shown in Figure 5.3. This figure illustrates the probability that a sensor's initial accuracy, state A, move to a different state in time t. Let's assume that $t_{max} = 5$ years (i.e., lifespan of the sensor is 5 years), then at $t = 0$, the probability that the sensor is in state A is 1, while as $t \geq 0$ the probability that the sensor is still in state A decreases. Thus, the greater the value of t , the greater the probability that a sensor change to state B, C and F respectively. For $t \longrightarrow \infty$, the accuracy state F of the sensor has a probability of 1 (i.e., the sensor is in failure state) Mailund [2018].

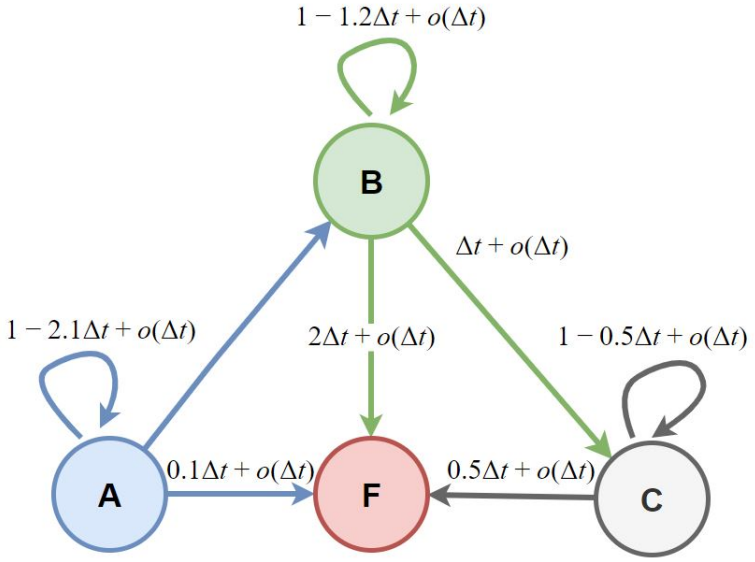

FIG. 5.2: Graphical representation of the Markov chain of the solution of the Kolmogorov differential equations of the proposed simulation.

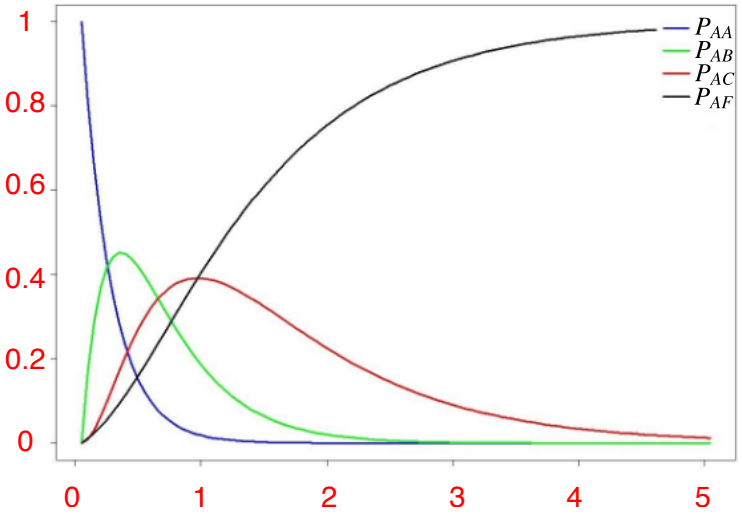

FIG. 5.3: Probability of a change in the accuracy state of the sensors from their starting point to the finish of their lifespan (x-axis is the time period in years and y-axis is the probability).

5.4.3 General description of the simulation

To test the proposed model we have chosen a building. At the time the sensors measured the temperature, the thermostat of the building showed 22°C. A mesh was used to place the sensors in the surface (Figure 5.1). With the help of laser levels, the sensors were placed vertically one in every different room. A total of 25 IoT nodes were deployed. The type of the sensor that was deployed was a combination of the ESP8266 microcontroller in its commercial version "ESP-01" and a DHT11 temperature and humidity sensor. The sum of both allows us for greater flexibility

when collecting data and adaptability to the case study, since the DHT11 sensor is designed for indoor spaces (it has an operating range of 0°C to 50°C) according to its datasheet [http://www.micropik.com/PDF/dht11.pdf]. The microcontroller collects data from this sensor through the onewire protocol and communicates it to the environment via Wi-Fi using HTTP standards and GET/POST requests. The ESP-IDF programming environment provided by the manufacturer of the microcontroller, was used to programme the device.

The sensors had been collecting data at 15 minute intervals, for an entire day. For the analysis we selected the data collected by the sensors in the following time interval $2018-08-20T09:00:00Z$ and ended on $2018-08-24T21:15:00Z$. A specific moment has been selected since the game that has been defined is static and not dynamic (i.e., it does not process the data in a temporal evolution). Below, a statistical summary of the measurements that were made with the sensors is presented in Table 5.6.

Tab. 5.3: Statistical table of measurements of the IoT nodes in case study III.

Timestamp start	**Total timestamp**	**Min temp**	**Max temp**	**Mean**	**Standard deviation**
2018-08-20T09:00	11:45:00Z	20.1 °C	24.6°C	22.91°C	0.71°C

5.4.4 Conclusion of the integration of the proposed future state prediction algorithm for the case study III

The proposed algorithm has been integrated in the proposed architecture. This algorithm makes possible to perform the future accuracy state prediction of the IoT devices. In this respect, the fact that the algorithm is able to predict the accuracy states is a good feedback to monitoring and control an IoT network. If you can predict when an IoT device will fail, you can change it before it sends false data. In this way, the monitoring of the IoT network is improved. In addition, the fact that it can be known with a high probability that IoT device will not be accurate allows to increase the robusted of the IoT network.

5.5 Case study IV: IoT network slicing and data quality algorithm

5.5.1 Introduction

Several reports indicate that commercial and residential buildings account for about 35% of total use of energy in the United States and Europe [Efficiency, 2009, Lapillonne et al., 2012]. Consequently, buildings are noted for being the biggest contributor of final power consumption, followed by industry and transport. Due to the potentially large energy savings that can be achieved through optimized use of energy in buildings, they have become one of the main targets for reducing global consumption. Despite being great consumers, private and public smart buildings have not exploited fully the full range of the energy efficiency chances presented to them. On the other hand, they are suffering a quite significant waste of energy which is in part caused by ineffective power systems such as cooling, lighting, heating and others (devices) [Terroso-Saenz et al., 2019], as well as the consumption behavior of inhabitants (behavior) [Coley et al., 2012] and poor insulation efficiency. Although in the first and third categories the implementation of heterogeneous energy measures is rather expensive, soft measures, that focus on changing the inhabitants' behavior, are cheap and at the same time very effective in reducing energy use [Vellei et al., 2016].

Heterogeneous data are not optimal inputs; as a result the precision of control algorithms is lower than if homogeneous data were used. Addressing the above mentioned inefficiencies as a result of lack of control algorithm which use heterogeneous data one could consider using homogeneous data and, in particular, homogeneous data collected by Internet of things networks. This novel method (i.e., homogeneous IoT data) which exists also at national level can also be used as a tool to realize the so called smart building. The installation of smart devices capable of using techniques or algorithms that convert the heterogeneous data collected by the IoT nodes into homogeneous data is going to improve the efficiency of the control algorithms since the input data is more reliable. The huge quantities of IoT information expected to be available from smart buildings in years to come should be examined to obtain information that can assist in obtaining, exposing and understanding the knowledge of buildings. On the other hand, resulting information may be able to assist in achieving significant energy efficiency strategies and actions in the target buildings. Therefore, it is essential to increase the effectiveness of information collection methods on IoT networks in smart buildings.

Motivated by the above observations, this case study have to test this novel technique for transforming heterogeneous data collected by IoT networks into homogeneous data

to improve the IoT network efficiency in control and monitoring. In this work the effectiveness of this technique will be tested with a data quality algorithm, which increases data confidence and detects wrong data. Compared with the existing results, the advantages of the proposed approach are summarized as follows.

1. By combining graph theory and clustering techniques together with the heterogeneous data collected by IoT networks, this thesis proposes a novel model to transform heterogeneous data in homogeneous data separated by layers. The proposed technique has the advantages of improving the effectiveness of the algorithm which have these data as input since our new model keep the clusters.

2. By using the IoT network slicing on virtual layer technique, algorithms are applied on homogeneous data. This way, there aren't malfunction due to the use of wrong data. The data quality algorithm that is optimized in this thesis has significantly improved its efficiency since the algorithm takes into account the neighborhoods but also the clusters (i.e., multiplex's layers).

In this regard it's necessary to develop this new algorithm which transform heterogeneous data into homogeneous data. To this end, we design a method to improve the IoT monitoring and control algorithm, focusing in our new proposed algorithm. This case study, is designed to test this new method. To prove the efficiency of this method, we test our data quality algorithm before applying this technique and then, after applying the IoT slicing method.

5.5.2 Experimental setup

In order to verify the efficiency of the IoT slicing with the data quality algorithm. The experiment was conducted in the 2^{nd} R+D+i smart building of the University of Salamanca which is shown in Fig. 5.1. By conducting the case study in this flood with the same technical characteristic, the proposed data quality algorithm can be validated in a one-day time period. We don't consider outdoor temperature. This experiment took place in January 28, 2019 after several simulations to test its efficiency.

5.5.3 General description of the simulation

To test the proposed model, we have chosen a smart building. At the time the IoT nodes measured the temperature. A mesh was used to place the sensors on the surface with the help of laser levels, the IoT nodes were placed vertically one in every section of the building. A total of 25 IoT nodes were deployed. The smart building where the

case study was deployed is shown in Fig. 5.1.

The type of sensor deployed in the building was a combination of the ESP8266 microcontroller in its commercial version "ESP-01" and a DHT22 temperature and humidity IoT node. The sum of both allows us for greater flexibility when collecting data and adaptability to the case study, since the DHT22 sensor is designed for indoor spaces (it has an operating range of 0°C to 50°C) according to its datasheet. The microcontroller obtains data from this sensor through the onewire protocol and communicates it to the environment via Wi-Fi using HTTP standards and GET/POST requests. The ESP-IDF programming environment provided by the manufacturer of the microcontroller, was used to programme the device.

The temperature sensor had been collecting data at 5 minute intervals, for 6 hours in the same day. For the analysis we selected the data collected by the sensors in the following time interval 2018-12-10T08:30:00Z and ended on 2018-12-10T14:30:00Z. To test the efficiency of the control algorithm a disturbance has been introduced in the temperature of smart building (our process) at 1 hour intervals to simulate the random behavior of people's thermostat use (i.e., a group of people could select different temperatures in their office thermostats). These disturbances have been introduced by in the members of our research group who had no consensus on which temperatures to introduce, so these temperatures can be considered pseudo-random. Below, a statistical summary of the measurements collected by the sensors is presented in Table 5.6.

Tab. 5.4: Statistical table of measurements of the IoT nodes in case study IV.

Timestamp start	**Total timestamp**	**Min temp**	**Max temp**	**Mean**	**Standard deviation**
2019-28-01T09:00	06:00:00Z	20.4 °C	24.7°C	22.8°C	0.87°C

5.5.4 Conclusion of the integration of the proposed IoT slicing technique for the case study IV

The integration of the IoT slicing method with the proposed architecture allows the system to improve the control of the smart building temperature. In addition, the use of this new method to transform heterogeneous data into homogeneous data allows the proposed method to improve the energy saving, because the temperature control is optimized.

Thanks to the use of the proposed method, the development of a system to model a new energy optimization in smart buildings. This allow us to optimize the energy saving in

smart building without the technical characteristics or the topology of the smart building influencing the functioning of the temperature control algorithms.

5.6 Case study V: Improving robustness in IoT networks

5.6.1 Introduction

In real-world network control systems (NCS), failures often occur in components of the systems (e.g. filters, actuators, sensors and controllers) mainly due to complex and difficult working as well as restricted network capacities. It has been reported that the frequency of faults it may be determined through fault detection techniques that have received substantial research assistance, while detailed data on faults can be collected through the error evaluation procedures, which provides a necessary precondition for further fault tolerant monitoring [Basin et al., 2015, Gao et al., 2015a,b]. With the increasing demands for safety, reliability, economic efficiency and service life of NCCS, the problem of failure sensing and evaluation has been extensively has been analysed up to now, and many findings have been reported in recent scientific literature [Li et al., 2017b, Qiu et al., 2018, Shahnazari and Mhaskar, 2018]. In particular, the issue of error identification in non-linear processes has explored in Samuel and Cao's paper using kernel component analysis techniques depth estimation [Samuel and Cao, 2016]. Furthermore, the paramenters of practical NCSs may vary over time as a result of the rapid complexity system and, consequently, the use of failure predictors for time management systems has drawn in considerable interest in the investigation (see, eg, related work [Casado-Vara et al., 2018c, Dong et al., 2016, Li et al., 2017a, Ren et al., 2017]). For instance, in the article of Dong *et al.* [Dong et al., 2016], regarding the merged effects of non-linearity, branched faults and fading channels, estimates of time-varying failures have suggested for finite horizon stochastic systems. Control system in real world, the appearance of failures in a system is inevitable due to the complexity of the system's architecture, long-term unexpected work changes in the surrounding outside neighbourhood. System faults can lead to degradation of the system or instability [Fattahi and Afshar, 2018, Tao et al., 2013]. Thus, it is necessary to conduct research on fault-tolerant control (FTC), which has gained a lot of attention in last decades. In general, FTC methods and technologies are split into active and passive approaches. Passive FTC is a technique that does not modify parameters and the structure in the controller Dong and Yang [2015]. Using this technique, the fixed controller is intended for the predetermined group of failures. However, the disadvantage is that the performance of the entire system can no longer be guaranteed if faults occur outside the default fault set. In contrast to the passive one, the active control algorithms performs fault compression setting the controller is switched online when malfunctions occur. Many approaches with active algorithms have introduced, such as sliding pattern approaches [Alaayed et al., 2013, Alwi and Edwards, 2008], observer-based methods [Zhang et al., 2009], multiple-model

method Boškovic and Mehra [2002], learning methods [Zhang et al., 2004] and adaptive compensation methods. Including, compensation in adaptive control method have been extensively used for the settlement of unidentified faults [Jin, 2016, Li and Yang, 2012]. However, there is one general limitation on the finite time existing outcomes that suggests that settlement time depends on starting parameters [Calvo-Rolle et al., 2015, Manuel Vilar-Martinez et al., 2014]. In other words, the time of convergence of the system changes depending on the starting parameters. On the ground, the starting parameters may be uncertain. IIn such instances, the above-mentioned finite time control method is unable to guarantee that the system reaches the required efficiency within a predefined and accurate time [Casteleiro-Roca et al., 2015, Sánchez Fernández et al., 2016]. Some results of nonlinear systems with fixed time control have been obtained [Casado-Vara et al., 2019b, El Bahja et al., 2014, Garcia et al., 2014, Quintian Pardo et al., 2012] based on other methods. However, as far as the authors' knowledge, few results focus on the use of prediction of accuracy states to improve the control signal. In this thesis, we intend to cover the research gaps in the field of monitoring and control of continuous-time networked systems with multiple IoT devices. Our aim is to present an improved control algorithm which will achieve the maximum allowable efficiency in predictive maintenance. A unified continuous-time hybrid control system model is presented together with a data quality and false data detection algorithm and a feedback control algorithm for predicting the accuracy state of the IoT sensor. The output of the data quality algorithm is the input of the predictive feedback control algorithm.

5.6.2 Case study setup

This case study supposes that the IoT nodes (i.e., temperature sensor) can undergo 4 accuracy states throughout their useful life (A=high accuracy, B=accurate, C=low accuracy, F=failure). The probability that a sensor in state A at instant t shift to state F in the time interval $(t, t + \Delta t)$ is $0.1\Delta t + o(\Delta t)$, if it is in state B it is $0.2\Delta t + o(\Delta t)$ and if it is in state C it is $0.5\Delta t + o(\Delta t)$. In this simulation we assume that the time during which the sensors remain in state A is an exponential time of $\lambda = 2.1$ in state A and $\mu = 1.2$ in state B.

From A in a time interval $(t, t + \Delta t)$ the sensor can pass to F with probability $0.1\Delta t + o(\Delta t)$. If ξ is the time the sensor stays at A, you have it:

$$P(\xi > t + \Delta t | \xi > t) = \frac{e^{-2.1(t+\Delta t)}}{e^{-2.1t}} = e^{-2.1\Delta t} = 1 - 2.1\Delta t + o(\Delta t) \tag{5.7}$$

Therefore, the (5.7) is the probability of remaining in state A at instant t_{i+1} if it was in A at instant t_i. Then, the probability of shifting to B between t and $t + \Delta t$ is

$$1 - ((1 - 2.1\Delta t + o(\Delta t)) - (0.1\Delta t + o(\Delta t))) = 2\Delta t + o(\Delta t) \tag{5.8}$$

In the successive stages we finally reach to a calculation in which the transition matrix is between t and $t + \Delta t$, as shown in Table 6.1.

	A	**B**	**C**	**F**
A	$1 - 2.1\Delta t + o(\Delta t)$	$2\Delta t + o(\Delta t)$	$o(\Delta t)$	$0.1\Delta t + o(\Delta t)$
B	0	$1 - 1.2\Delta t + o(\Delta t)$	$\Delta t + o(\Delta t)$	$0.2\Delta t + o(\Delta t)$
C	0	0	$1 - 0.5\Delta t + o(\Delta t)$	$0.5\Delta t + o(\Delta t)$
F	0	0	0	1

TAB. 5.5: In this case study, we have assumed that state F is absorbent. That is, for the sensor to move from F to any other state, it needs to be repaired by a maintenance worker.

Thus, the derivative of the matrix in the zero is:

$$P'(0) = \begin{pmatrix} -2.1 & 2 & 0 & 0.1 \\ 0 & -1.2 & 1 & 0.2 \\ 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.9}$$

which may be expressed using the Jordan matrix form for the whole period of time t as follows:

$$P(t) = \frac{1}{0.504} \begin{pmatrix} 1 & 1 & 2 & 1 \\ 1 & 0.8 & 0.9 & 0 \\ 1 & 0.56 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & e^{-0.5t} & & \\ & & e^{-1.2t} & \\ & & & e^{-2.1t} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0.504 \\ 0 & 0 & 0.9 & -0.9 \\ 0 & 0.56 & -0.8 & 0.24 \\ 0.504 & -1.12 & 0.7 & -0.084 \end{pmatrix} \tag{5.10}$$

For example, the term $p_{AF}(t)$ represents the probability that a sensor that begins its useful life at stage A, will function incorrectly at time t, so:

$$P(\text{Life span} \leq t) = p_{AF} = 1 - \frac{0.9}{0.504}e^{-0.5t} + \frac{0.48}{0.504}e^{-1.2t} - \frac{0.084}{0.504}e^{-2.1t} \tag{5.11}$$

Figure 5.2 shows a graphical representation of the Markov chain. Probabilities of changes in the accuracy states of the sensors are shown in Table 6.1. The instance simulation presented in this section demonstrates that sensors in any of the precision states (i.e., A,B,C) can move to a state of malfunction(F).In this figure, however, the sensors only

degrade by one state; from state A to state B and from state B to state C. This is so, since in this example we assume that the sensor from any of its precision states can fail, while, we assume that a high accuracy sensor (A), has to go through the precise state (B) before moving to the low accuracy state (C).

Given the Markov chain used for this simulation with transition matrix given by the eq. (5.10), the stationary paths given by the probabilities of change of precision state of the sensors are shown in Fig. 5.3. This figure illustrates the probability that a sensor's initial accuracy, state A, will shift to a different state in time t. Let's assume that $t_{max} = 5$ years (i.e., lifespan of the sensor is 5 years), then at $t = 0$, the probability that the sensor remain in state A is 1, while at $t \geq 0$ this probability decreases. Thus, the greater the value of t , the greater the probability that a sensor change to state B, C and F respectively. For $t \longrightarrow \infty$, the accuracy state F of the sensor has a probability of 1 (i.e., the sensor is in failure state) [Mailund, 2018].

5.6.3 General description of the experiment

To test the proposed model, we have chosen a supermarket. At the time the IoT nodes measured the temperature, the thermostat of the supermarket showed 23°C. A mesh was used to place the sensors on the surface of the ground floor (Fig. 5.4). With the help of laser levels, the IoT nodes were placed vertically one in every section of the supermarket. A total of 25 IoT nodes were deployed.

The type of sensor deployed in the supermarket was a combination of the ESP8266 microcontroller in its commercial version "ESP-01" and a DHT22 temperature and humidity sensor (Fig. 5.4). The sum of both allows us for greater flexibility when collecting data and adaptability to the case study, since the DHT22 sensor is designed for indoor spaces (it has an operating range of 0°C to 50°C) according to its datasheet [`http://www.micropik.com/PDF/dht22.pdf`]. The microcontroller obtains data from this sensor through the onewire protocol and communicates it to the environment via Wi-Fi using HTTP standards and GET/POST requests. The ESP-IDF programming environment provided by the manufacturer of the microcontroller, was used to programme the device.

The temperature sensors had been collecting data at 15 minute intervals, for an entire day. For the analysis we selected the data collected by the sensors in a time interval that began on 2018-10-17T09:00:00Z and ended on 2018-10-17T21:15:00Z. A specific moment was selected since the game that has been defined is static and not

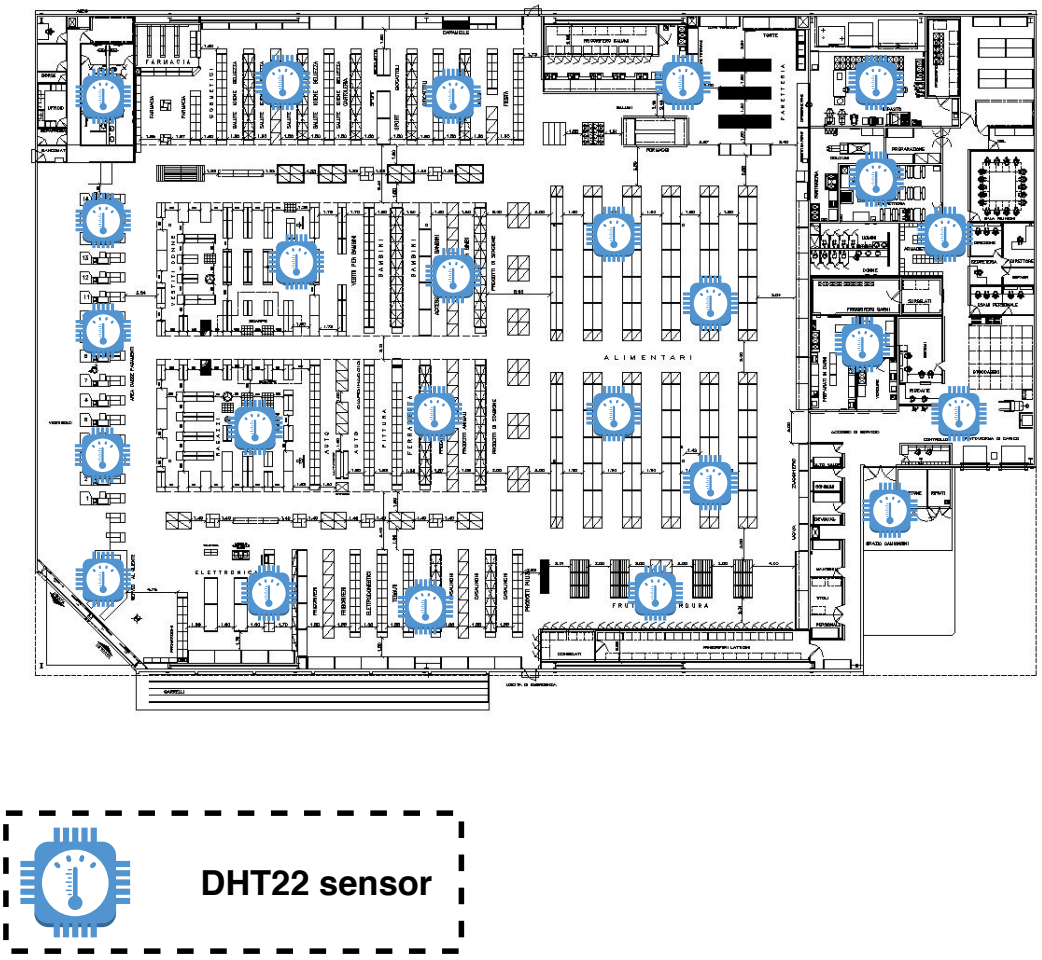

FIG. 5.4: Map of the supermarket showing the distribution of the sensors, a sample of an indoor surface for testing our model and a sample of a sensor.

dynamic (i.e., it does not process the data in a temporal evolution). Table 5.6 provides a statistical summary of the measurements that were collected by the sensors.

TAB. 5.6: Statistical table of measurements of the IoT nodes in case study V.

Timestamp start	**Total timestamp**	**Min temp**	**Max temp**	**Mean**	**Standard deviation**
2018-10-17T09:00	11:45:00Z	20.1°C	24.6°C	22.91°C	0.71°C

In this experiment we have considered the next time interval $(t, t + \Delta t)$: $\frac{1}{365 \cdot 5}$ (i.e., a day). To validate the model we applied the accuracy state prediction model to the data collected by the sensors placed in the supermarket.

5.6.4 Conclusion of the integration of the proposed fault-tolerant control algorithm for the case study V

This case study is oriented to the main use of the fault-tolerant algorithm and is focusing on the field of energy saving by improving the robustness of the IoT network. Advances in this field can have a major economic impact on the smart building funds and the environment.

Moreover, this case study seeks to have an impact on the smart building environment, reducing energy consumption by improving the robustness of the IoT network and improving the monitoring and control of the smart building.

In addition, fault-tolerant algorithm development has enabled the preventive management of the smart building IoT network. This way, the IoT network is operating longer without collecting false data by faulty devices.

5.7 Conclusions

The designed architecture and the algorithms have allowed to deploy several case studies in different ways to prove their efficiency in smart building environments. A total of five case studies have been described, which have made it possible to evaluate the design of the proposed system in stages. The proposed case studies lead to the following conclusions:

- The algorithms have been adapted to the needs and have development, being able to be easily modified when the requirements and objectives of each case study require it.
- The proposed algorithms provides the advantage of having access to data collected in the case studies, these data can also be used to test future algorithm or improvements of these algorithms. These data are going to allow to simulate the same temperature conditions in the application of new techniques and technologies.
- The algorithms that have been designed are auto-adaptable to any IoT network without the need to adjust the topology of the IoT network or the characteristics of the IoT devices so that they can operate.

Chapter 6

Results

VNiVERSiDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Chapter 6

Results

6.1 Introduction

This chapter presents the results obtained form the case studies described earlier in chapter 5 and the conclusions drawn form each of them. This chapter shows the results of the case studies that have been designed to prove the effectiveness of the proposal of this thesis. Each case study presents the results associated with that part of the proposal we have made in this thesis.

These case studies have been designed with the help of researchers from the BISITE group to test the efficiency of the algorithms designed in this Doctoral thesis. Each of them is a situation designed to test each of the algorithms designed. In the real case studies, the IoT devices were placed and the IoT network was put into operation with the help of experts from the BISITE group. The data collected in these simulations were used to feed the algorithms and the results were extracted from them.

The rest of the chapter is organized as follows: section 6.2 shows the first case study results about the efficiency of the data routing protocol in IoT architectures, section 6.3 presents the results of the data quality case study. Section 6.4 shows the future accuracy IoT devices states prediction results. Section 6.5 presents the IoT network slicing technique results focusing on improve the data quality algorithm. Section 6.6 shows the algorithm that improve the robustness of the IoT network. Finally, section6.7 presents the conclusions.

6.2 Case study I: Data routing in IoT architectures

6.2.1 Introduction

A simulation environment has been designed using data from a real smart building and collected data regarding temperature conditions for energy consumption. This will allow to verify that the architecture and our novel algorithms adapts to the smart building context before implemented it in a real-world IoT network. In this way, we are going to acquire knowledge the subsequent cases studies it will be necessary to continue working on these parts to include new techniques and technologies.

6.2.2 Results

In this section, two simulation cases are considered to illustrate the effectiveness of the novel adaptive control algorithm and the faster data search using hashmap.

6.2.2.1 Simulation 1: Queuing adaptive control algorithm

This simulation of the adaptive control algorithm is shown in Table 6.1. This table shows the results of the simulations of the adaptive control algorithm for 20 stages with 4 IoT nodes that randomly send blocks to the block queuing system.

Values taken by the three different $\mu_{controller}$, $\mu_{reference}$ and $\mu_{adtative}$ can be found in Fig. 6.1a. The $\mu_{controller}$ parameter is the signal that the controller sends to the block queuing system (process), while the $\mu_{reference}$ parameter receives the signal that comes out of the process and sends it as feedback to the adaptive control. In this way it can self-correct the error in these parameters. In this figure we can find the fact that the parameters μ that represent the capacity of miners are always equal or greater than the entry of blocks in the queue (i.e., λ). This is important because it is proof that the control algorithm prevents the queue from collapsing, and if this happens, the algorithm makes the queue work correctly again in the next step of the algorithm. Fig. 6.1b verifies that the $\mu_{adaptive}$ is lower bounded by the value of λ. So, the μ of the process will always be higher than the λ. Our findings suggest that the $\mu_{process} = \mu_{adaptive}$ self-adapts to the $\mu_{process}$ allowing the block queuing system to recover from saturated conditions, and otherwise, the $\mu_{process}$ self-adapts to the $\mu_{process}$ in order to ensure that $\rho < 1$. This way we check that the control algorithm works correctly. This is because the capacity of the server is always greater than the number of blocks that enter the

TAB. 6.1: Result of the 20 step numerical simulation demonstrating how the adaptive control algorithm works.

Step No	$\gamma_{1,2,3,4}$	λ_5	ρ	Q^*	$u(\lambda, \mu, Q^*)$	$\mu_{controller}$	$\mu_{reference}$	$\mu_{adaptive}$
1	$\{8, 4, 6, 2\}$	20	$\frac{2}{3}$	20	0	20	24	22
2	$\{6, 9, 10, 7\}$	32	$\frac{32}{22}$	0	-1	22	29	33
3	$\{4, 8, 7, 5\}$	24	$\frac{24}{33}$	19	1	33	29	31
4	$\{5, 4, 9, 1\}$	19	$\frac{19}{31}$	21	0	20	20	20
5	$\{7, 5, 3, 4\}$	19	$\frac{19}{20}$	7	1	20	25	22
6	$\{5, 4, 8, 3\}$	20	$\frac{20}{22}$	10	1	22	26	24
7	$\{8, 6, 9, 2\}$	25	$\frac{25}{24}$	0	-1	24	26	25
8	$\{10, 7, 1, 6\}$	24	$\frac{24}{25}$	7	1	25	32	28
9	$\{9, 8, 6, 5\}$	28	$\frac{28}{28}$	0	-1	28	26	35
10	$\{6, 4, 1, 2\}$	13	$\frac{13}{35}$	22	0	17	16	16
11	$\{5, 9, 1, 10\}$	25	$\frac{25}{16}$	0	-1	16	38	27
12	$\{2, 5, 3, 4\}$	14	$\frac{14}{27}$	10	1	16	19	17
13	$\{6, 3, 5, 10\}$	24	$\frac{24}{17}$	0	-1	16	27	25
14	$\{5, 6, 2, 1\}$	14	$\frac{14}{25}$	19	0	16	16	16
15	$\{8, 5, 7, 1\}$	21	$\frac{21}{16}$	0	-1	16	20	24
16	$\{7, 5, 10, 1\}$	23	$\frac{23}{24}$	7	1	24	31	27
17	$\{5, 4, 3, 10\}$	22	$\frac{22}{27}$	15	1	27	27	27
18	$\{4, 7, 6, 9\}$	26	$\frac{26}{27}$	7	1	27	34	30
19	$\{3, 4, 2, 10\}$	19	$\frac{19}{30}$	20	0	19	20	19
20	$\{9, 6, 5, 3\}$	23	$\frac{23}{19}$	0	-1	19	18	24

queue.

Fig. 6.1c shows the parameters ρ and the value of the controller function u. This figure shows that both variables are closely related. So, when the variable $\rho \geq 1$, $u = 1$ (i.e., if the dashed red line is over the green line, the system is saturated). In other words, the controller informs the system that once the block queuing system is saturated, it has to auto-adapt the system's ρ to overcome this state. This is achieved by increasing the value of $\mu_{process}$, as $\rho = \frac{\lambda}{\mu}$. Thus, if $\mu > \lambda$ the system is no longer saturated and returns to its stationary state. This graph shows how the control algorithm reacts when the tail collapses. This prevents the miners' network from collapsing and avoids delays in the monitoring and control of the IoT network. In addition, Fig. 6.1d shows the behavior of the Q^* parameter in relation to u. Since u is directly related to ρ as presented in Fig. 6.1c, Q^* is also directly related to ρ. Eleven $u = -1$ ($\rho \geq 1$), the controller detects that the system is saturated and therefore the optimal value of the blocks that have to be on hold for the miners' network to mine blocks is zero (i.e., $Q^* = 0$).

6.2.2.2 Simulation 2: Hashmap search

This simulation has been done using HDFS to store the data of the IoT nodes. The objective of this simulation is to validate that it takes less time to search the data using

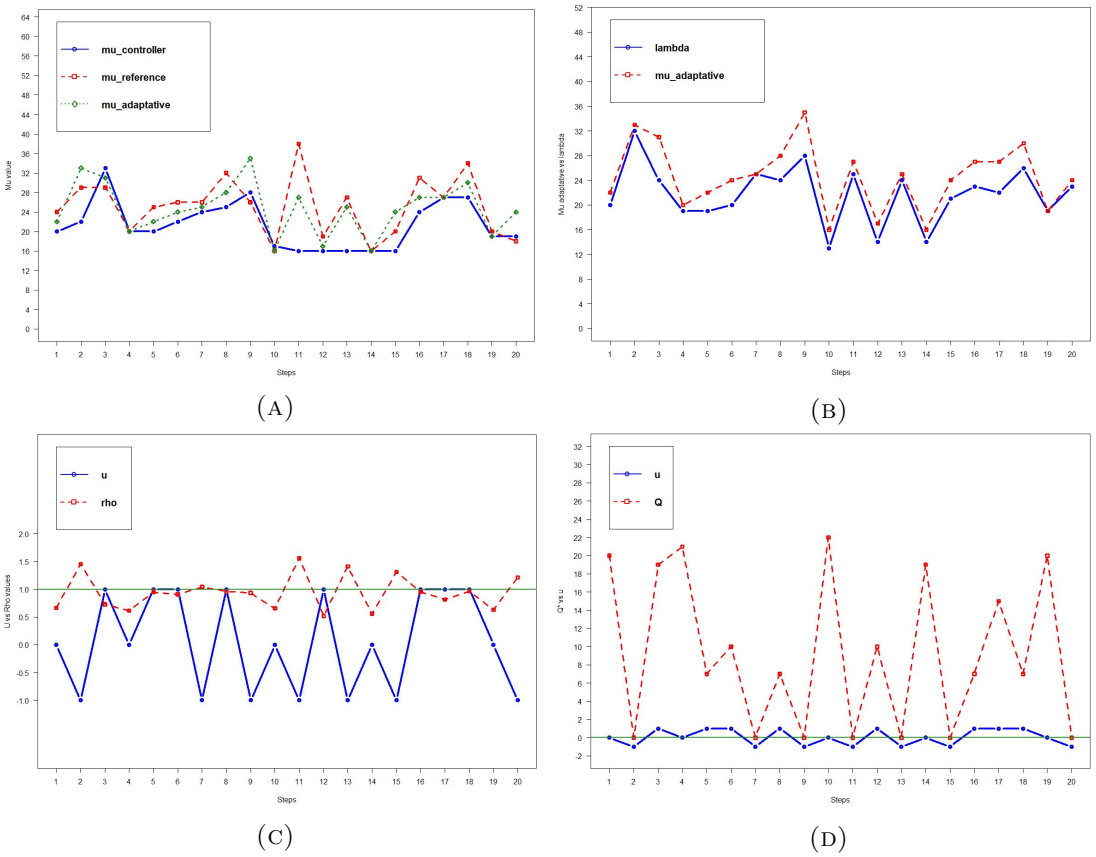

FIG. 6.1: **Adaptive control algorithm measurements.** (a) The $\mu_{controller}$ (solid blue line), the $\mu_{reference}$ (dashed red line) and the $\mu_{adaptive}$ (dashed green line) generated by simulation 1. In this figure we can find the $\mu_{controller}$ and $\mu_{reference}$ errors and how the $\mu_{adaptive}$ self-correct these errors. (b) The $\lambda_{miners'\ network}$ (solid blue line) and the $\mu_{adaptive}$ (dashed red line) generated by simulation 1. In the figure we can find how the $\mu_{adaptive} \geq \lambda_{miners'\ network}$ in all the steps of the control adaptive algorithm. (c) The controller function u values (solid blue line) and the $\rho_{miners'\ network}$ (dashed red line) generated by simulation 1. In this figure we can find a relationship between u and $\rho_{miners'\ network}$. In this way, once $\rho_{miners'\ network} \geq 1$ then $u = -1$. The saturated bound (solid green line) shows the top bound for the system to not be saturated. (d) The u controller function value (solid blue line) and the Q^* value (dashed red line) generated by simulation 1. Since $u = -1$ then $Q^* = 0$.

a hashmap than it is in HDFS. For the HDFS simulation, the data of IoT nodes were randomly generated and stored in HDFS. While for the hashmap simulation, the sensor ID, timestamp and query were stored in the hashmap for subsequent data search in the HDFS. In this way, it is easy to monitor and control IoT nodes, by locating the data through their sensor ID field, or their timestamp field. The search via hashmap is very fast (its computational complexity is $O(1)$). The different simulations that have been generated are shown in Fig. 6.2. The size of the data has been increased to see how both search systems behave. In all the simulations, search via hashmap proved to be faster.

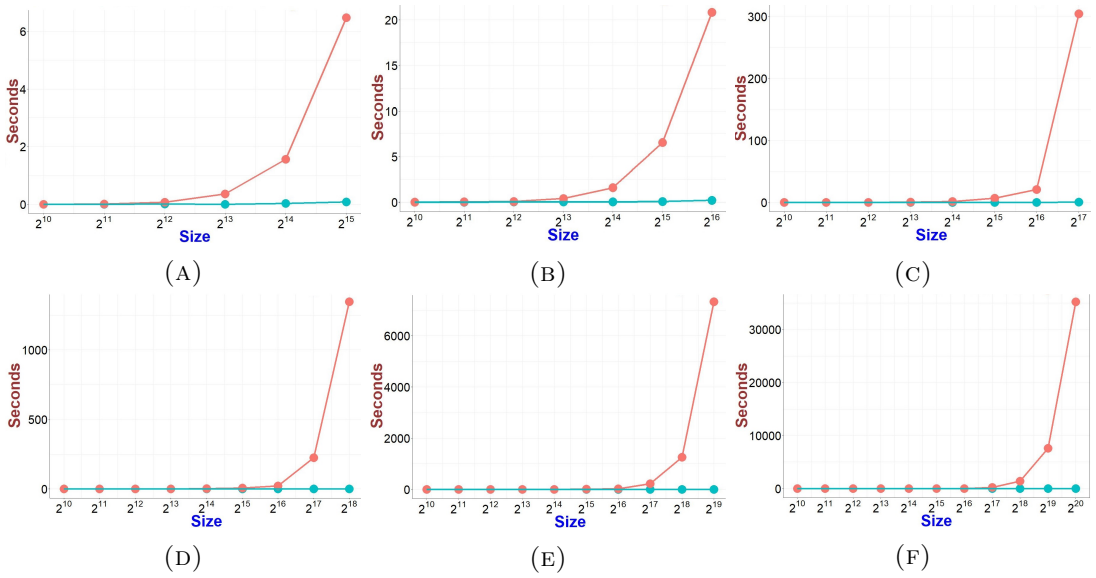

FIG. 6.2: **Data search speed up using hashmaps.** The time (in seconds) of the search process using hashmaps (solid blue line) and the time (in seconds) of the search process with big data technologies (solid red line) generated by the simulation with different size values.

6.2.3 Conclusion and future work

This case study has addressed the block control flow problem between IoT devices and blockchain. By using queuing theory, adaptive controller is developed to achieve the optimal block number to improve the mining process efficiency. In this way, stability of the miners' network is improved since the adaptive controller ensures that the network is not saturated. On the other hand, a new model to speed up the searches by using hashmaps is proposed in this case study. Thus, a new architecture to merge these new contributions is proposed to improve IoT platforms. The effectiveness of the proposed approach is supported by simulations.

The extensions to systems with more general structure are an interesting topic for future research. The present case study leaves open several future lines of research. The first of these consist of an experimental system to building optimized energy saving smart buildings. The next objective is to try to achieve better routing ways in the IoT network architecture. This implies a new routing algorithm to improve the current one.

6.3 Case study II: Data quality based on consensus

6.3.1 Introduction

This case study has shown the data quality algorithm that allows to improve monitoring and control of the IoT network for energy saving in smart buildings. Therefore, the next step is to deploy the system in real environment with dynamic conditions. To ensure a valid evaluation of the system and to demonstrate that it constitutes and advancement in energy saving that is not caused by smart building characteristic or topology. The following subsection details the setting in which the validation of the proposed algorithm has been performed.

6.3.2 Results

As highlighted throughout the thesis, this proposal focuses on the application of a distributed and self-organized GT game to the temperature data collected by a IoT network from an indoor surface. Our main goal is to analyze the temperature data provided by each sensor and to verify the quality of this data assuming an error of 1%. Our game, was applied to a matrix containing the temperatures collected in a time $t = t_0$ by the IoT network. However here we only focus on a time t_0(i.e. we considered a static system). One of the main benefits of our game is that it is distributed and self-organized, which is a great advantage when dealing with data obtained by a IoT network.

In figure 6.3, the initial temperature is shown in the first image, and in the rest of the images the iterations of the game until the GE is reached. In the successive images the temperature clusters are being formed, this can be observed by the changes in the colour gradient. It can also be seen that some areas with inaccurate temperatures are smoothly self-corrected on the basis of the temperatures in their environment. This is the intended process, since the game is executing its iterations depending on the environment surrounding the sensor. This makes sense because the temperature of the sensor will be similar to the average temperature of the environment in which it is located. Also, we show the evolution of the temperature on the surface. To this end, we have represented the temperature on the z axis to facilitate visualization in the form of a surface. The first image shows the temperature measured by the sensors. The game is applied iteratively to every image until it reaches the game equilibrium. Notice that the proposed game transforms the temperature data according to its environment. If we consider this as a knowledge data discovery (KDD) process, the temperature collected

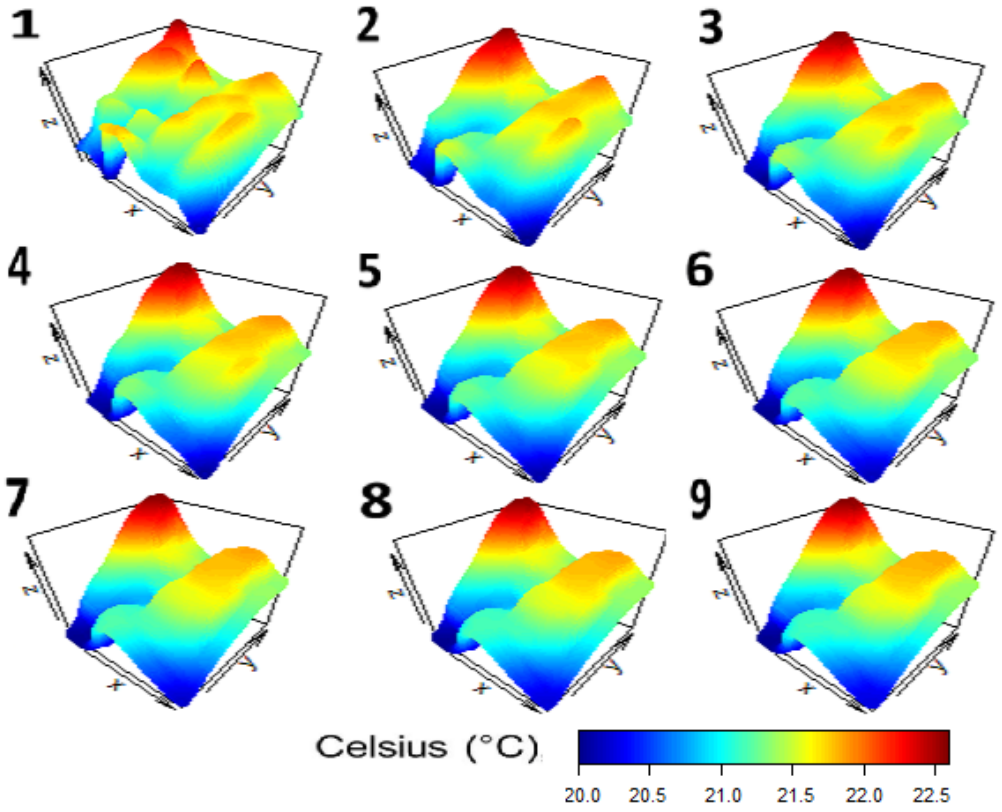

FIG. 6.3: Evolution of temperatures surface over the different steps in the game until reaching GE.

by the sensors would be the target data (TD), the game performs the pre-processing and data transformation to the TD simultaneously. Resulting in some transformed data, which, when the game reaches the NE are the surface temperatures in the GE. In addition, since the game is self-organized and distributed, it can be applied to the data in the ETL (extract, transform and load) process without having to go through a central node first.

Once the GE is reached in the game, it can be seen that temperature clusters are formed. In Fig. 6.4, we compare our method with K-means. The clusters can be observed, in the top row with the initial temperatures and in the bottom row, the temperatures in the GE of the game. The method is more robust and adaptable than some unsupervised classification methods, such as K-means. One of the main advantages of the developed algorithm over K-means, is that before searching the clusters in the data K-means needs to have the number of clusters. On the other hand, our algorithm can find the number of clusters by itself. This is a remarkable advantage in the exploratory analysis of data, since in an unsupervised way it will give us very valuable information about the zones with similar temperatures. This data can be used to find the number of optimal centroids for the unsupervised classifiers (e.g., K-means).

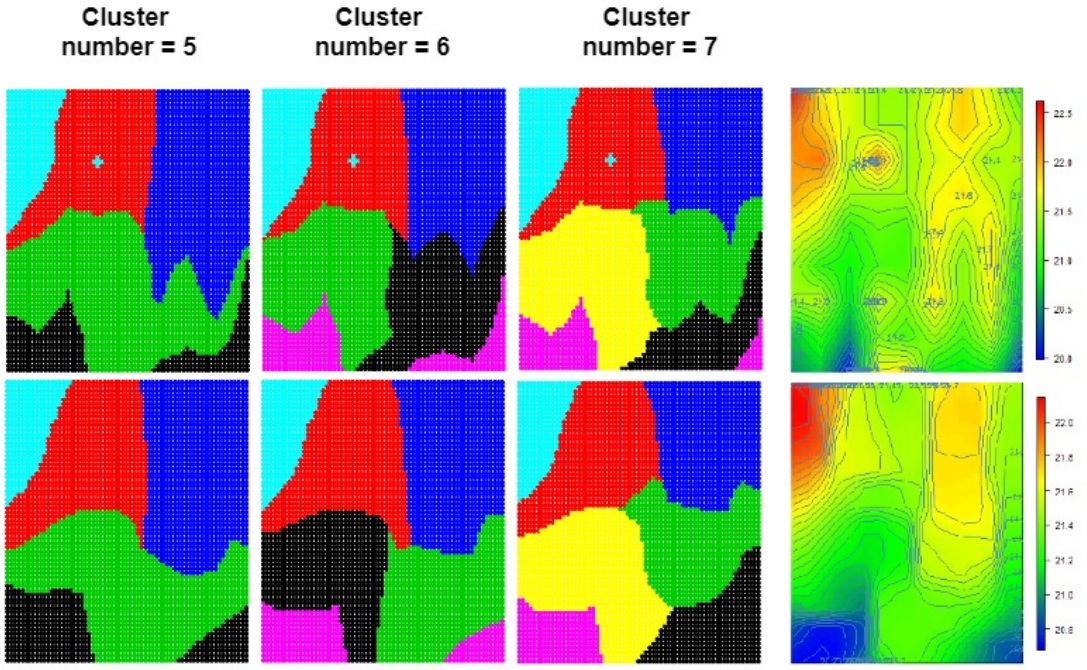

FIG. 6.4: Data before applying the algorithm (top row) and after applying the algorithm (bottom row). K-means is used to validate the effectiveness of the proposed algorithm for self-correcting the data. In the first column k-means with 5 clusters is used, and in the second and third columns k-means with 6 and 7 clusters respectively is used.

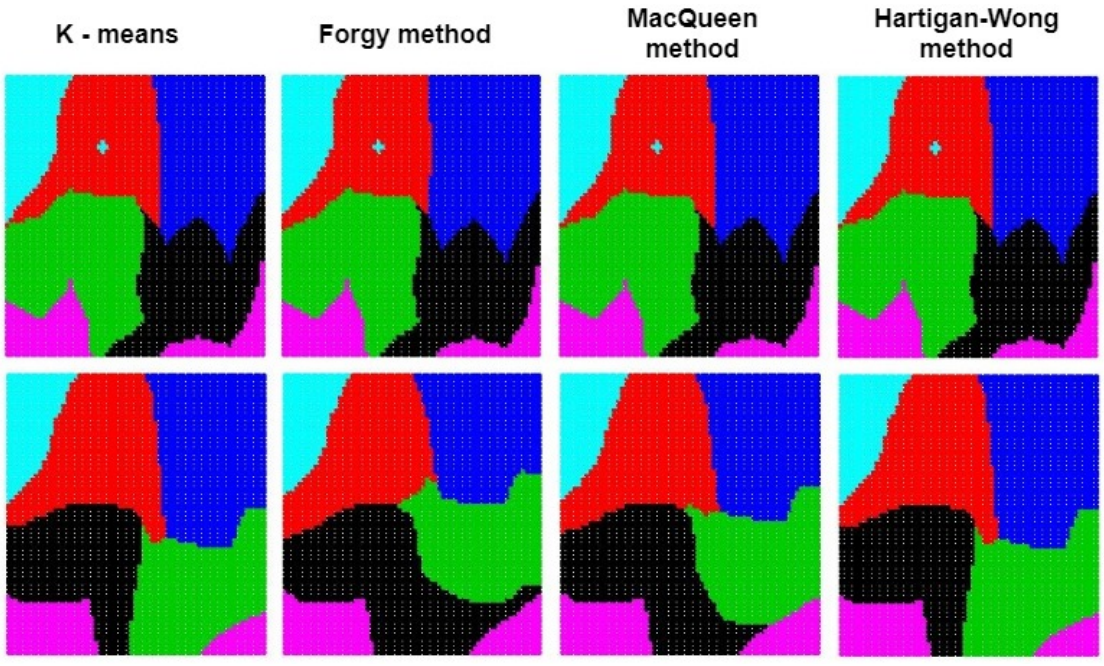

FIG. 6.5: Data before applying the algorithm (top row) and after applying the algorithm (bottom row). Four clustering methods are used to validate that the proposed algorithm self-corrects data. In addition, the figure shows how the proposed algorithm interacts with different clustering methods.

In Fig. 6.6 we show an execution of K-means for the initial temperatures (first row), and for the temperatures changed by the game into equilibrium (second row). The first three images in each row are the execution of k-means on the temperatures, varying the number of centroids between 5, 6 and 7. A representation of a heat map with the isothermal lines is added to each of the rows. Once the GE has been reached in the game, the isotherms show that the number of suitable clusters is 6. In fact, looking at the execution of k-means for 6 centroids, it was found that the formation of clusters done by K-means is very similar to that on the isotherm map. Another result is that anomalous temperatures are self-corrected (on the basis of their environment) when the game is executed. We can notice in the first image in figure 6.3 that some temperature values get smoothed as the game iterations increase (both maximum and minimum). In this way, similar temperatures are achieved, which is correct, since the appearance of critical points in the data that differ significantly from their environment suggests that they may not be reliable. In Fig. 6.4 it is shown how the game self-corrects these points.

In figure 6.5 a panel of two rows of images are presented. In the upper row are the images of different clustering methods applied to the temperatures collected by the IoT network. In the lower row, the same clustering methods are applied but the data has already been self-corrected by the algorithm proposed in this case study. In figure 6.5, different clustering methods have been used to see how the proposed algorithm behaves compared to different clustering methods. In order, the clustering methods used for this comparison are: 1) K-means. 2) Forgy clustering method. 3) MacQueen clustering method. 4) Hartigan-Wong clustering method. In the application of these clustering methods we have used the optimal number of clusters for the data we have used in the experiment, the optimal number of clusters is 6. In the 4 columns you can see how the proposed algorithm auto-corrects the data without making substantial changes in the clusters that make up each of the chosen methods. However, it can be noted that the proposed algorithm auto-corrects the inaccurate data and validates the data for later use. One of the novelties of our proposed algorithm is that it auto-corrects inaccurate data and minimally modifies clusters. In fact, in Fig. 6.5 this novelty has been validated with 4 different cluster methods.

Assume $\epsilon = 0.01$ for the defined fuzzy convergence criterion, in Fig. 6.7 we show the evolution of the elements that converge in each of the stages of the game, starting from $n \geq 2$. With the convergence speed equal to or greater than the order of $O(x^6)$ we can state that in a few iterations of the game the fuzzy convergence criterion is reached. In our case there are 9 iterations until convergence.

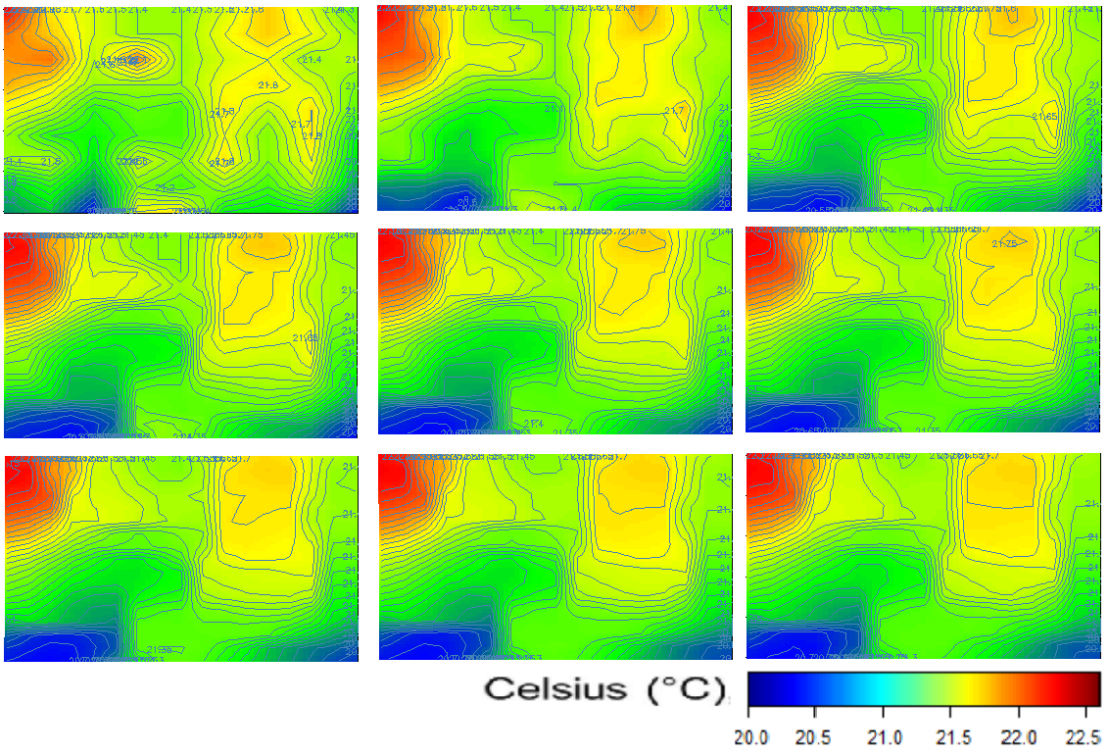

FIG. 6.6: Evolution of surface temperatures in the different steps of the game until reaching GE in a map of isothermal lines. The images are ordered in rows from left to right. The first image shows the initial temperatures, and in the remaining images we see each of the game iterations until GE is reached.

The self-correction of the temperature values collected by the IoT network is a notable outcome of our research. This self-correction is performed on the basis of the average temperature in the surrounding environment. IoT network will provide a temperature matrix for a time $t = t_0$. The advantage of self-correction is that the data provided by the IoT network will be reliable; there will be no anomalous data, such as temperature peaks (maximum or minimum) which differ significantly from the temperature in the neighbourhood. This helps to eliminate (to a large extent) the possible thermal noise introduced by the sensors that are functioning incorrectly or inaccurately, facilitating the task of monitoring the temperature of a surface.

TAB. 6.2: Table with the different permitted errors and % noise before and after the application of the game.

Allowed error (º Celsius)	**Sensors with noise at the beginning (%)**	**Sensors with noise at the end (%)**
0.01	15.23	1.04
0.05	47.06	13.25
0.1	70.59	24.22
0.2	92.74	48.10
0.25	96.20	60.90
0.3	98.27	71.98

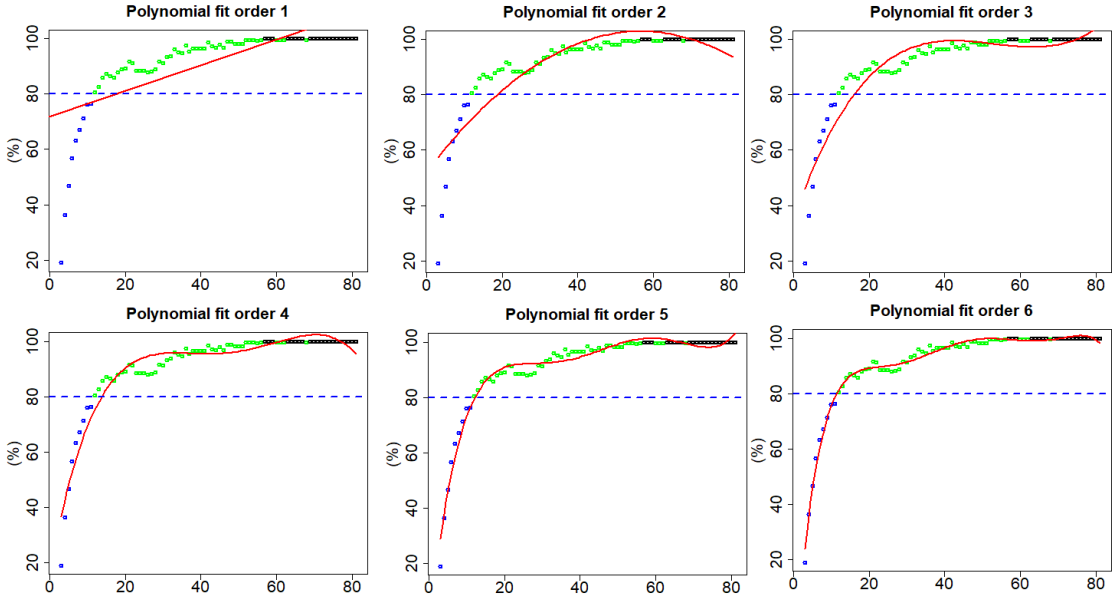

FIG. 6.7: Panel with a linear, quadratic and cubic adjustment of % of temperature matrix elements convergent with the evolution of the iterations of the game. A blue dashed line has been place on each of these images marking the defined fuzzy convergence zone. Points below the convergence zone are also blue, the points above the area of convergence, and black when the convergence is 100 %. It can be noticed that the game converges to the 9 iterations for the fixed ϵ. In addition, with the help of the regression lines highlight the degree of convergence is of the order of $O(x^6)$.

The proposed algorithm provides an excellent transformation in the ETL process, in the data flow itself, we can apply our work as a transformation included in the ETL process for the generation of new temperature data, that is already self-corrected and ready to be used. A large part of the relative thermal noise brought by the data arriving from the sensor, is eliminated (noise is created when sensors are defective or not accurate). Figure 6.8 shows the number of sensors (in percentages) with thermal noise for each iteration of the game. It can be noticed that by switching the precision of the sensors from $0.01^{\underline{o}}$ to $0.3^{\underline{o}}$, the obtained results are quite different. However, if changes to $0.01^{\underline{o}}$, 15 % of the sensors had thermal noise, and in a few (< 10) iterations the noise was reduced to less than 5 %. When the allowed relative error was increased the percentage of sensors having some thermal noise also increased. For example, with $0.3^{\underline{o}}$ of relative error, 95 % of the sensors had thermal noise and as the iterations increase it was reduced to less than 75 %. However, at some point the noise began to plateau. These sensors will continue to have some noise for the chosen error (Table 6.2).

There are two other very useful applications of our work: 1) The identification of sensors which provide false data and the calibration of new sensors by injecting them into the IoT network. 2) The intelligent false data detection in a IoT network is an important issue, since it allows for the predictive maintenance of the IoT network, as well as a high level of data quality. Also, predictive maintenance allows to maintain the IoT network

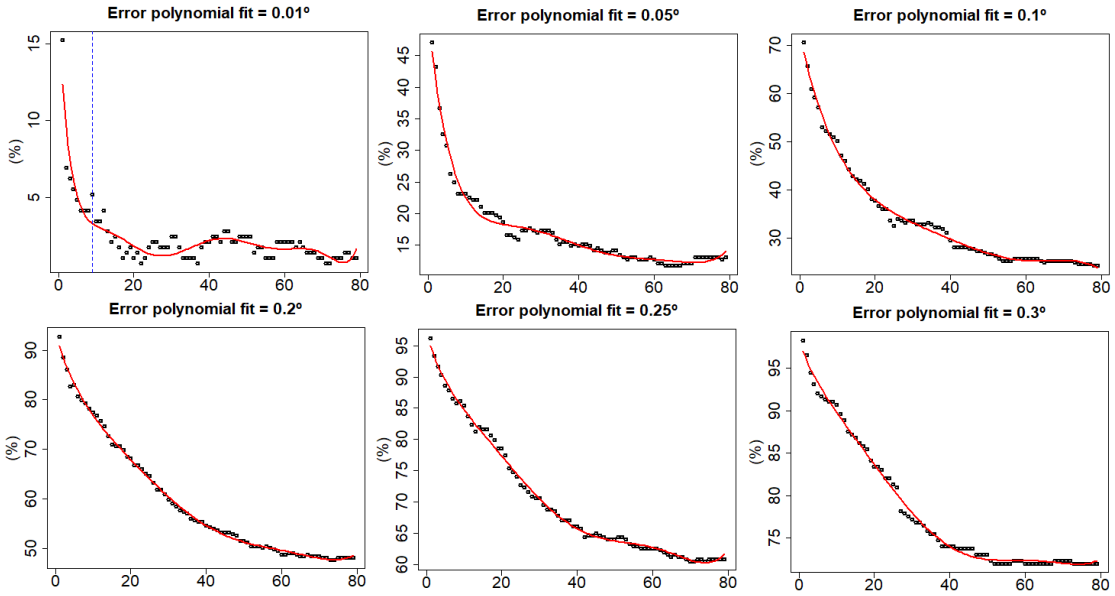

FIG. 6.8: Panel with the decrease of thermal noise in the evolution of the game with different margins of confidence from 0.01 degrees Celsius to 0.3 degrees Celsius. In the image panel the % noise in the temperature matrix is presented versus the number of iterations. In each of them, the permitted margin of error for the temperature collected by the sensors is varied. It can be seen that as the permitted margin of error increases, the thermal noise in the temperature matrix also increases.

functioning properly. Since the malfunctioning sensors are identified, the maintenance cost reduces significantly as the technician can focus on defective sensors only.

On the other hand, the level of confidence that the game provides to the data is very high, since it eliminates the defective and/or inaccurate sensors together with a large part of the noise. This provides us with high quality and fairly accurate data.

Another application of our work is the location of zones with similar temperature (clusters). This is very useful in the field of intelligent monitoring of buildings since hot and cold bulbs can be detected. Furthermore, heat leak can be identified with the proposed game and predictive measures can be taken (such as reinforcing the materials with thermal insulators or modifying the heating in the building in order to adjust the temperature in areas with heat leaks).

Another possible use of the game would be smart agriculture. For example, a crop field with a IoT network monitoring temperature, AI techniques can be applied to find very hot and cool areas, in order to decide if they need watering or some other type of treatment. This type of predictive maintenance can be very attractive for private industry.

After discussing the results and possible applications of our work, we stress that the major novelty of our research in the field of IoT network is that, we do not focus on

the IoT network itself, but on the treatment of the data collected by the IoT network. As described in the results section, the proposed game is distributed and self-organized. This allows to treat the data obtained from the IoT network in order to make them reliable. Identified clusters can discover defective or inaccurate sensors, which allows for predictive maintenance of the IoT network.

6.3.3 Conclusion and future work

This work proposes a distributed and self-organized cooperative algorithm using game theory. The algorithm has been applied to the data collected by a IoT network from an indoor surface. The main goal of the game is to make IoT network monitoring more robust through consensus temperature monitoring. Furthermore, the presented work achieves and ensures data quality, false data detection (i.e., inaccurate sensors) and temperature data optimization to improve energy efficiency in cooperative IoT network. The most significant results obtained in this work are listed below.

The game is distributed; a central node is not required to run. The game is also self-organized since IoT network nodes interact with each other to generate self-corrected data in relation of their surrounding environment. Unlike the classic unsupervised methods, our work does not require the number of clusters beforehand. Once the game reaches the GE, the resulting number of clusters can be used as input for the unsupervised classification analysis.

On the other hand, anomalous temperature values are corrected according to their surroundings, without modifying the temperature clusters. That is, the game self-corrects without changing the cluster structure that is obtained directly by the IoT network in the temperature matrix. Finally, with the established diffuse convergence criteria, the game converges in very few iterations (9) to the GE. Here we summarize the assumptions that have been made in the case study in order to reduce complexity and avoid confusion. The surface on which the sensors were placed was regular and known. This is a very significant limitation, since the target surface will not always be a regular surface. Moreover, we have assumed that the data is collected in a time $t = t_0$ and that the game is static. This limits our model since it does not evolve over time. Finally, only a Von Neumann neighbourhood was used with $r = 1$, so the environment is fairly small and could slightly affect the rate of convergence of the game.

Despite these limitations, our work provides a novel, distributed and self-organized algorithm, which allows to self-correct temperature data collected by the sensors according to their surrounding temperatures. We also address some interesting results and some quite promising industrial applications. Future lines of work will be focused

on extending the game to larger topological manifolds and we will study these manifolds dynamically. Future lines of research include the study of incidence of new factors affecting energy consumption, as well as the inclusion of a greater number of devices to collect these new data from the smart building.

6.4 Case study III: Future states prediction

6.4.1 Introduction

It is necessary to be able to check if the future accuracy states prediction algorithm allows energy saving. To perform a valid evaluation of the algorithm and to demonstrate that the algorithm allows for optimize monitoring and control of IoT networks in smart buildings; different simulations in the case study setting are chosen for validation. In this case study, the Markov chains in continuous time were incorporated for improve energy savings in smart buildings. The configuration performed in the case study is detailed below.

6.4.2 Results

In this simulation we have considered the next time interval $(t, t + \Delta t)$: $\frac{1}{365*5}$ (i.e., a day, 365 days in one year and the lifespan of the devices are 5 years) and five loops of the algorithm every day (i.e., 5 days of algorithm operation), starting from the time t=0 (i.e., 2018-08-20T09:00). We are going to apply the accuracy state prediction model for this first example with the data collected by the sensors placed in the building to validate the model.

6.4.3 Simulation 1: Three days prediction scenario

For this simulation the time interval $(t, t + \Delta t)$ used is 1 day. In this way, we predict the accuracy states of the sensors from their start point $t = 0$ to $t + \Delta t = \frac{1}{365*5}$ (i.e., one day). Since the useful life of the sensor is 5 years, it is expected to check with this simulation that the accuracy of the sensors will remain the same as when they were installed (i.e., $t = 0$). In this first loop example we detail the process of the control algorithm, while the other 2 steps, according to the scheme of the first loop are shown in a shorter way. The simulation input and output are shown in Table 6.3.

In this first loop of the prediction of the accuracy states does not change with respect to the initial accuracy states of the sensors. Since the time interval assumed $(t, t + \Delta t)$ is one day. Therefore, the degradation and accuracy loss of the sensors has been minimal. Therefore, as the coefficients of the matrix $T^{p(t)} \leq 9$ (0.5% $t_{max} = 9$ days) there is no need to replace any sensor. In Fig. 6.9 the outcomes of the predictive control algorithm are shown.

Input	First day	Second day	Third day
T_e	$\begin{bmatrix} 0.8 & 0.7 & 0.4 & 0.2 & 0.5 \\ 0.7 & 1.1 & 0.5 & 0.2 & 0.1 \\ 0.8 & 2.4 & 1.0 & 1.6 & 0.4 \\ 1 & 0.4 & 0.4 & 0.7 & 0.1 \\ 0.6 & 0.4 & 0.7 & 0.3 & 0.3 \end{bmatrix}$	$\begin{bmatrix} 0.6 & 0.8 & 1.5 & 0.4 & 1.1 \\ 0.1 & 0.3 & 0.2 & 0.1 & 0.1 \\ 0.4 & 1.9 & 1.0 & 0.6 & 0.1 \\ 1.6 & 0.4 & 0.4 & 1.1 & 0.5 \\ 1.9 & 1.1 & 0.2 & 0.1 & 1.5 \end{bmatrix}$	$\begin{bmatrix} 0.8 & 0.3 & 1.9 & 0.6 & 0.1 \\ 0.1 & 0.8 & 1.7 & 0.1 & 0.2 \\ 0.6 & 1.9 & 1.2 & 0.5 & 0.4 \\ 1.1 & 0.2 & 1.1 & 1.1 & 0.9 \\ 0.9 & 0.1 & 0.4 & 0.1 & 1.6 \end{bmatrix}$
Output	First day	Second day	Third day
T_p	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 1 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 3 & 3 & 1 & 0 \\ 3 & 0 & 1 & 2 & 1 \\ 2 & 1 & 0 & 0 & 2 \end{bmatrix}$

TAB. 6.3: Accuracy state prediction algorithm input and output for simulation 1.

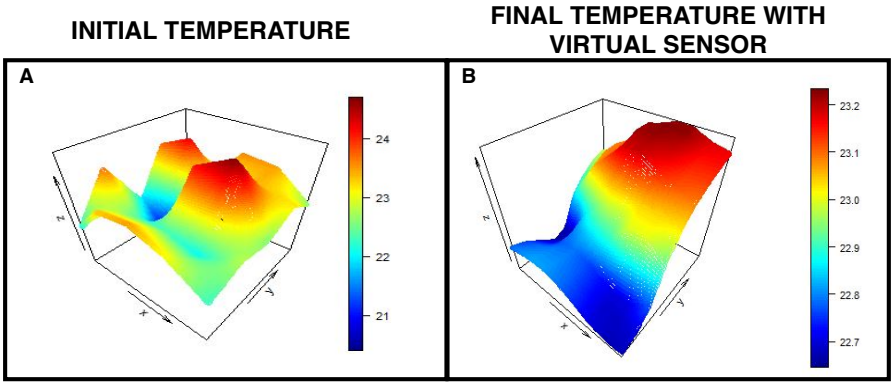

FIG. 6.9: Graphic representation of the matrix of initial temperatures of the first loop (A) and final temperatures of the process of the first loop (B). In the figure (A) can be found the temperatures collected by the IoT nodes. In addition, the measurements that the control algorithm will find as false data can be found. Figure (B) shows the final temperatures after the control algorithm is executed.

In this second loop you can see how the control algorithm adjusts the temperatures using the feedback function. Thus, in this second loop the controller does not detect any sensor in fault state. Therefore, the control algorithm has improved the precision of the sensors. In Fig. 6.10 the outcomes of the predictive control algorithm are shown.

In this last loop the algorithm monitors and controls the temperature of the building. Using the proposed model it is able to predict the accuracy states of the sensors in the future and based on these predictions it controls the temperature.In Fig. 6.11 the outcomes of the predictive control algorithm are shown.

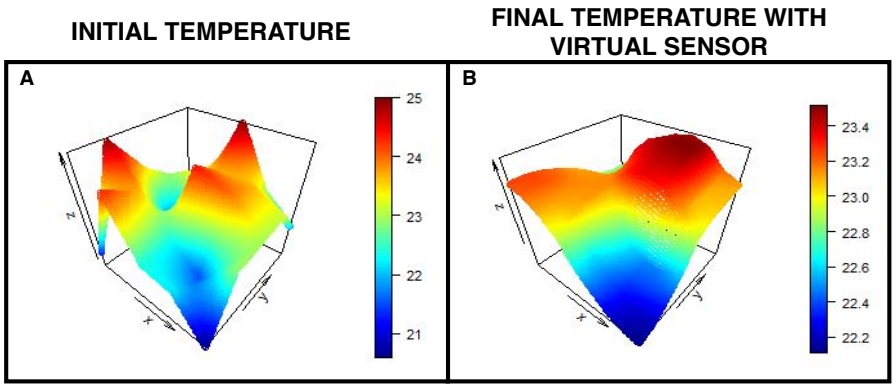

FIG. 6.10: Graphic representation of the matrix of initial temperatures of the second loop (A) and final temperatures of the process of the second loop (B). In the figure (A) can be found the temperatures collected by the IoT nodes. In addition, the measurements that the control algorithm will find as false data can be found. Figure (B) shows the final temperatures after the control algorithm is executed.

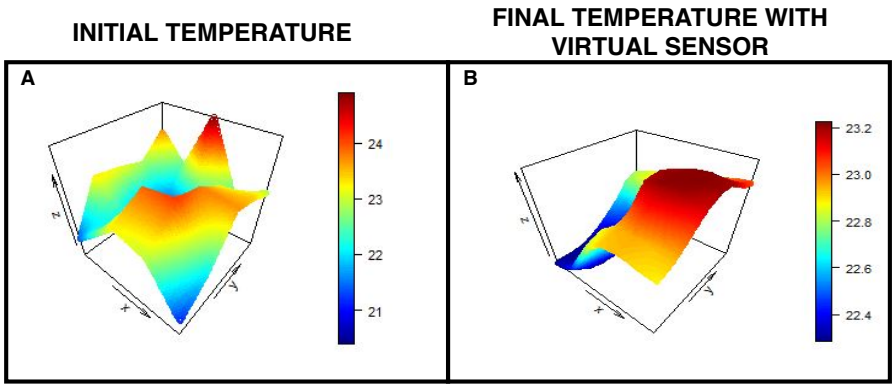

FIG. 6.11: Graphic representation of the matrix of initial temperatures of the third loop (A) and final temperatures of the process of the third loop (B). In the figure (A) can be found the temperatures collected by the IoT nodes. In addition, the measurements that the control algorithm will find as false data can be found. Figure (B) shows the final temperatures after the control algorithm is executed.

6.4.4 Simulation 2: IoT node in failure state scenario

In this simulation we assume that there are several IoT nodes in faulty state. The time interval $(t, t+\Delta t)$ used is 1 day. The simulation input and output are shown in Table 6.4.

In Fig. 6.16 it is possible to find the measurements of the initial temperature matrix T_i and the final temperatures given by the control algorithm T_f after detecting the sensors that are in failure and replacing them with virtual sensors. In Fig. 6.16. (A) can be found the 4 sensors that the algorithm found as false data. Then the control algorithm predicts the future accuracy states of all IoT nodes. In this way, the IoT nodes whose predicted state is in failure for several time intervals are replaced by an IoT virtual node. Thus, the control algorithm estimates the temperatures of these IoT virtual nodes with the data quality algorithm (see Appendix A). Finally, the

Input	faulty test simulation
T_e	$\begin{bmatrix} 6.15 & 0.94 & 1.61 & 1.35 & 4.01 \\ 0.32 & 1.53 & 2.34 & 0.52 & 0.41 \\ 0.59 & 1.97 & 3.78 & 0.13 & 0.02 \\ 1.03 & 0.31 & 0.34 & 0.37 & 1.74 \\ 1.01 & 0.23 & 0.03 & 0.79 & 4.21 \end{bmatrix}$
Output	Faulty test simulation
T_p	$\begin{bmatrix} 8 & 1 & 1 & 2 & 8 \\ 0 & 1 & 3 & 2 & 0 \\ 0 & 3 & 8 & 3 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 8 \end{bmatrix}$

TAB. 6.4: Accuracy state prediction algorithm input and output for simulation 2.

process returns the matrix of final temperatures improved by the control algorithm. In Fig. 6.16. (B) the final temperatures can be found. It is shown that the final temperatures given by the predictive control algorithm are homogeneous and without false data or malfunctioning IoT nodes.

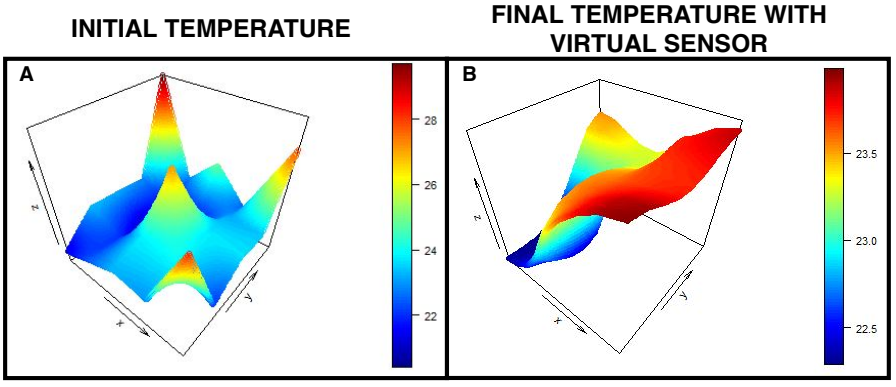

FIG. 6.12: Graphic representation of the matrix of initial temperatures (A) and final temperatures of the process (B). In the figure (A) can be found the temperatures collected by the IoT nodes. In addition, the measurements that the control algorithm will find as false data can be found. Figure (B) shows the final temperatures after the control algorithm is executed.

6.4.5 Conclusion and future work

This case study has addressed the problem of predictive control of accuracy in continuous-time NCSs. The feasibility of the proposed approach was verified with a case study in which the closed-loop system was modeled as a continuous-time feedback system with the Kolmogorov differential equations to predict the future accuracy states of the IoT nodes. Through a newly constructed feedback control algorithm and a data

quality algorithm (see Apendix A), an improved control system has been created. It allows to derive a smart building's maximum allowable energy efficiency such that the resulting closed-loop system improves the control of an IoT network. A numerical example illustrates the efficiency of our model. However, in many real scenarios, the ability to detect an imprecise or malfunctioning IoT node from a hot (cold) spot is limited. In a future work, we will try to solve this problem with artificial intelligence. In future research we will examine the possibility of designing an architecture that would incorporate this algorithm.

6.5 Case study IV: IoT network slicing and data quality algorithm

6.5.1 Introduction

A real case study has been designed using data from a smart building IoT network. This will allow to verify this new method adapts for several algorithm for control smart building environment values. In this way, we will acquire knowledge of the strong point of the characteristic and topology of the smart buildings, as well as the parts that need to be improved.

6.5.2 Results

The first thing we do is build the graph of the IoT network and apply the clustering algorithm. In Fig. 6.13 one can find both graphs. After applying the clustering algorithm, it can be observed how 4 clusters are formed. Therefore, the multiplex in Fig. 6.14 has got 4 layers. Thus, data is homogeneous. In this case study, non-connected graphs are formed on all layers of the multiplex using the data collected from the smart building. This is a common situation, and the algorithm self-corrects this by virtualizing all the required nodes for the graphs to be connected.

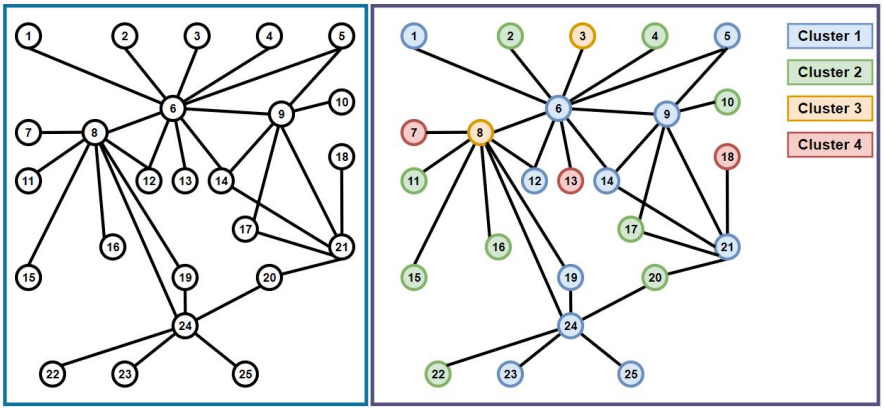

FIG. 6.13: Graph of the IoT nodes in the smart building and colored graph by clusters.

The results obtained from the comparison of the use of the control algorithm are presented in the Fig. 6.15. In this figure you can find the result of the application of the data quality algorithm on heterogeneous data. In this case the data quality algorithm smoothes the data collected by the IoT network so that the data quality algorithm understood that the heterogeneous data were homogeneous. For this reason,

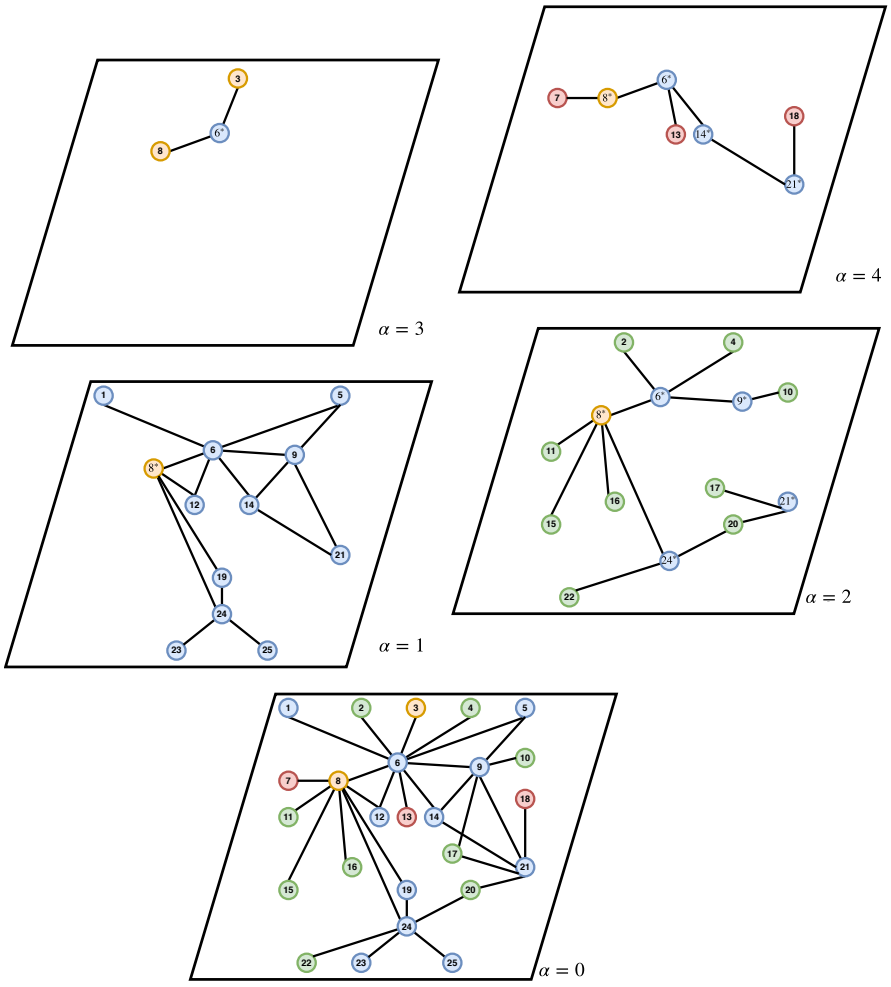

FIG. 6.14: Multiplex with the 4 layers. In every layer we have to virtualize some IoT nodes to the graph will be connected.

the algorithm detects some temperatures collected by the IoT network and considers them outliers. Then the self-correcting algorithm corrects these values and the output of the data quality algorithm are homogeneous temperature data but this would not increase the energy efficiency as it does not use the information from the clusters to distinguish this information. On the other hand, using the technique proposed in this case study, through which heterogeneous data are transformed into homogeneous data with clustering techniques and complex networks, the energy efficiency of the IoT network is increased since the data collected by the IoT network by applying the control algorithm only acts for the homogeneous areas and thus maintains the temperature clusters as output of the data quality algorithm.

6.5.3 Conclusion and future work

This case study has investigated the inaccuracy problem of IoT network algorithms using heterogeneous input data. Through the introduction of a complex network and clustering

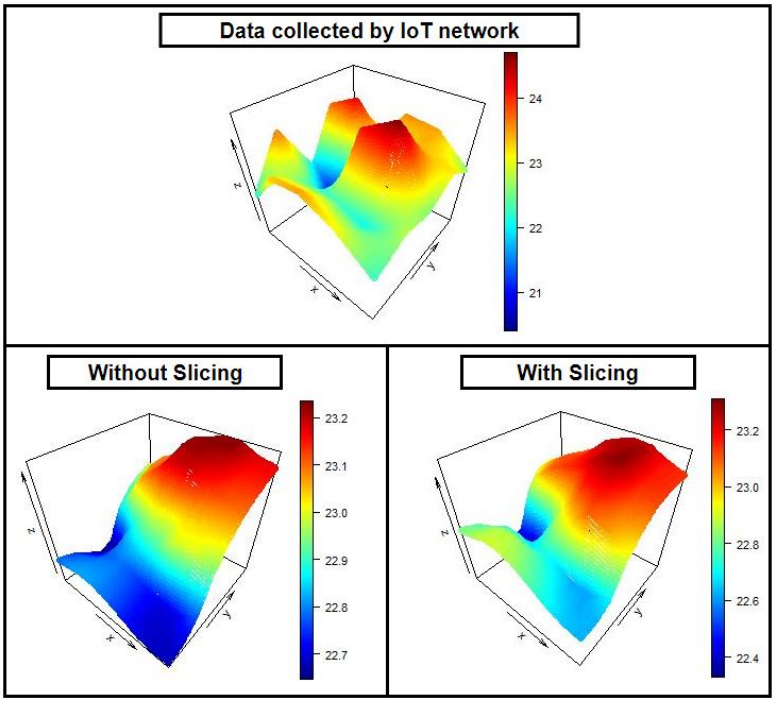

FIG. 6.15: The results obtained by the data quality algorithm are compared using the technique proposed in this case study and without using the technique.

techniques, these heterogeneous data can be virtualized into segmented virtual layers considering the clusters in order to transform heterogeneous data into homogeneous data that optimizes the operation of the algorithms. Using the virtual segmentation technique provided by our new method, the algorithms guaranteeing an optimized performance considering the different areas of the topology of the IoT network. Finally, this case study result is given to demonstrate the efficacy of the proposed Iot slicing method. Future work will be concentrated on application of this novel method to several Iot control algorithms with heterogeneous data input.

6.6 Case study V: Improving robustness of IoT networks

6.6.1 Introduction

This case study has shown that the fault-tolerant algorithm allows for energy savings in a smart building. To comply a valid evaluation of this algorithm and to demonstrate that it constitute and advancement in energy savings that is caused by the fault-tolerant algorithm that improves the robustness of the Iot network. The following subsection details the setting in which the validation of the proposed algorithm has been performed.

6.6.2 Results

In this section we introduce the case study and the results achieved during the entire experiments. The control algorithm retrieves the data gathered from the IoT nodes and corrects them automatically. In addition, if the controller forecasts that an IoT node will be faulty, it will generate a virtual temperature sensor to keep the IoT network reliable. This improves the efficiency of monitoring and controlling the IoT network.

6.6.3 Case study results

In this case study, we are going to consider a system with external disturbances. In the measurement time 4 disturbances have been introduced randomly into the system. In Fig. 6.16, the temperature of the controlled system is shown. In this figure one can find that the desired temperature son 23° Celsius. It should be pointed out that, in the smart buiding, disturbances not only come from our case study setup but also arise from a wide range of things (people activities, malfunctioning heating system, etc.). In this case study, we only consider the disturbances introduces by our team.

From Fig. 6.17, it can be shown that the predicted accuracy states of the IoT nodes of the smart building. In this figure we can see the state prediction performance after the disturbance introduction. During the settling time, the state prediction module output is showing that the IoT nodes are in low accuracy state or even in faulty state.

In Fig. 6.18, state trajectories of predicted and collected data are shown. In the figure one can notice that when perturbations are introduced the predicted temperature is slightly higher than that collected by the IoT nodes. But when the control algorithm reaches the equilibrium, then the difference between the two trajectories is minimal.

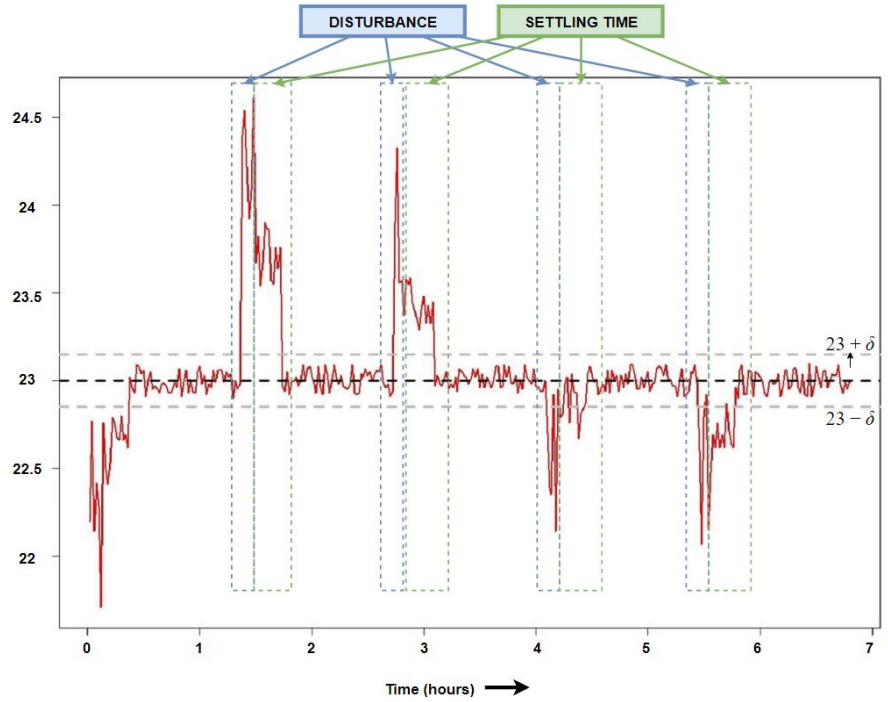

FIG. 6.16: Temperature collected by the smart building's IoT nodes during the time of the case study. In blue, the 4 disturbances of the office temperature of the smart building are indicated and in green, the settling time is indicated, which takes the temperature to reach the desired temperature after the disturbances.

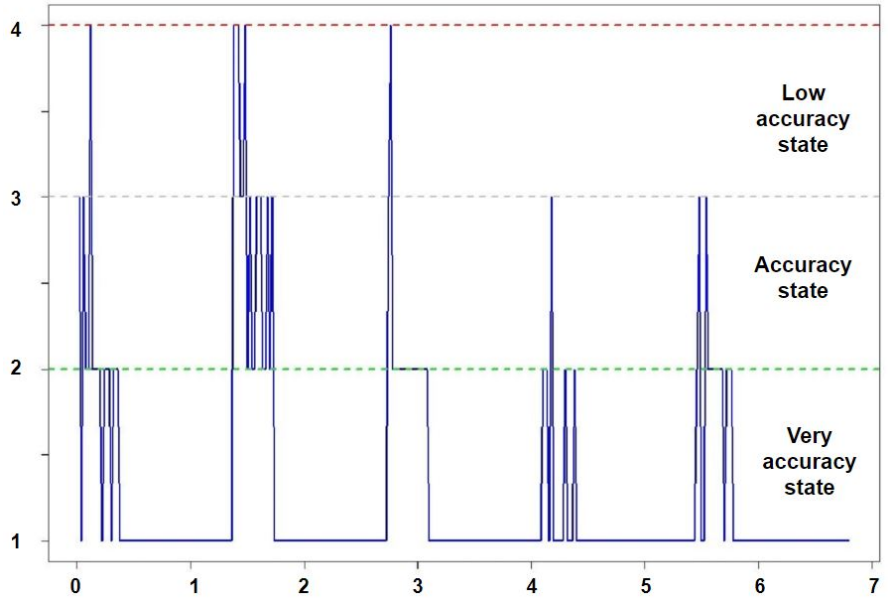

FIG. 6.17: Prediction trajectories of the accuracy states of IoT nodes.

Tracking predicted error $\Delta S_{sp}(t)$ is presented in Fig. 6.19, and the tracking error convergence is shown after the disturbances introduced in the system.

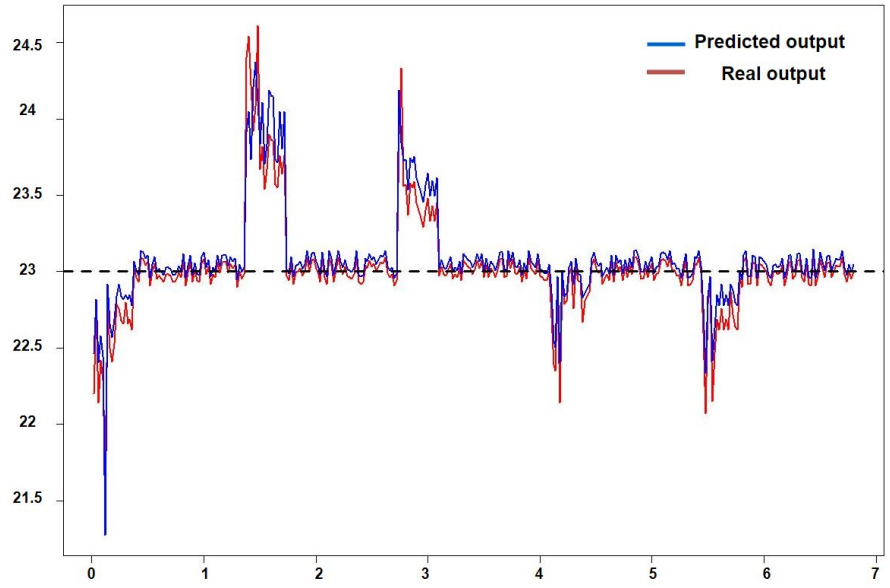

FIG. 6.18: State trajectories comparison of the predicted output (solid blue) with the real output (red) of the control algorithm.

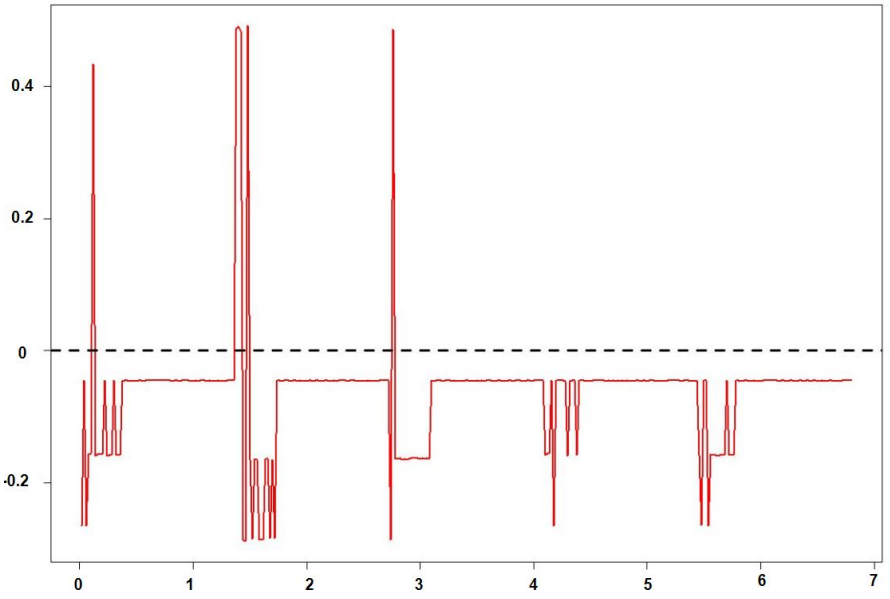

FIG. 6.19: System tracking predicted error $\Delta S_{sp}(t)$.

6.6.4 Conclusion and future work

In this case study, the issue of fault-tolerant robust adaptive control with accuracy future states prediction is investigated for IoT temperature networks with external disturbances. By introducing the data quality and state prediction modules, within the apriori given user-defined commanded temperature bounds the tracking error can be guaranteed. To minimize tracking error, this novel algorithm provides this new modules can optimize the robustness of the IoT network since it has improved its fault-tolerant performance. Using the accuracy state estimation data provided by the

adaptive algorithm, this algorithm can predict the output temperature and compare with the collected temperature from the IoT network. Closed-loop outputs are bounded by the commanded temperature interval even we introduces some disturbances the settling time is very small. Finally, a case study is proposed to prove the efficiency of the developed algorithm. Future work will be concentrated on robust adaptive algorithm for nonlinear system with time delay and external uncertaines disturbances.

6.7 Conclusions

This section shows the functionality of the improved architecture and new algorithms and draws conclusions from its implementation in the case studies. A system has been developed based on the improved architecture and algorithms model, which can deployed in several case studies to achieve the objective of reducing energy consumption. In this dissertation, the effectiveness of the proposed model has been evaluated in the five case studies proposed. The deployed case studies have shown the suitability of the proposed techniques for energy efficiency in dynamic IoT network. The overall conclusions are:

- First case study has addressed the block control routing problem between IoT devices and blockchain. By using queuing theory, adaptive controller is developed to achieve the optimal block number to improve the mining process efficiency. Also, this algorithm allows to optimizes the management of queues in IoT architectures by increasing energy savings. Moreover, it improves database searches which increases the effectiveness of IoT network monitoring.

- The second case study shows a distributed and self-organized cooperative algorithm using game theory. This algorithm allows to find the wrong data and self-correct them. Moreover, in this case study it has been shown that the algorithm reduces the "thermal noise" introduced by the malfunctioning IoT devices. Finally, this algorithm allows to create an error scale that we have used to increase the accuracy of IoT networks.

- Third case study has addressed the problem of predictive control of accuracy in continuous-time NCSs. The feasibility of the proposed approach was verified in this case study. This algorithm allows to estimate the accuracy states of the IoT devices in future time, therefore, this algorithm allows to improve the predictive maintenance of the IoT network.

- Fourth case study has investigated the inaccuracy problem of IoT network algorithms using heterogeneous input data. Through the introduction of a

complex network and clustering techniques, these heterogeneous data allow to be virtualized into segmented virtual layers considering the clusters in order to transform heterogeneous data into homogeneous data that optimizes the operation of the algorithms. This new method allow to improve the monitoring and control task in IoT networks.

- Fifth case study, the issue of fault-tolerant robust adaptive control with accuracy future states prediction is investigated for IoT temperature networks with external disturbances. By introducing the data quality and state prediction modules, within the apriori given user-defined commanded temperature bounds the tracking error can be guaranteed. This novel algorithm optimize the robustness of the IoT network since it has improved its fault-tolerant performance and minimize tracking error of the IoT control algorithms.

Chapter 7

Conclusions and future work

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Chapter 7

Conclusions and future work

7.1 Introduction

This chapter describes how the different objectives defined in this research have been achieved in order to validate and evaluate the starting hypothesis: "*The starting hypothesis of the present research work is formalized in that it is possible to improve or optimize some of the current techniques and technologies for the monitoring and control of IoT dynamical networks in smart buildings.*"

This research work has presented an IoT middle-ware layer architecture and modular algorithms (integrated in the layer) for the optimization of energy consumption in smart building, monitored and controlled by IoT networks. The main novelty is that the designed optimization algorithms has led to more efficient processes in terms of energy consumption for the monitoring and control of IoT networks in smart buildings. This proposal allows to optimize the energy consumption in smart buildings without having to make large economic investments because the system designed can be integrated into any IoT architecture regardless of the characteristics or topology of the smart building. The work presented has included the validation of the developed techniques through the deployment of the operational proposal by simulation with realistic conditions and experimentation in real environments.

The rest of the chapter is organized as follows: section 7.2 presents the final conclusions of this thesis, section 7.3 shows the state of the art contributions and, finally, section 7.4 presents the future work of this research work.

7.2 Final conclusions

This section presents the final conclusions of this Doctoral thesis. These conclusions are drawn from all the research work carried out in this thesis on the monitoring and control

of dynamic IoT networks to optimize the use of energy in smart buildings. The main contributions achieved in this Doctoral Thesis are the following:

- It has carried out an analysis of the concrete problems of monitoring and control in IoT networks in smart buildings detected in the revision of the state of the art, tackling the existing deficiencies related to energy savings. This has served to establish the set of initial requirements for the design and development of the solution.
- It has studied the techniques and technologies used in the monitoring and control in IoT networks in smart buildings that has allowed the design of a solution attending to the optimization of the energetic consumption. These requirements have led to the development of a new technique to transform heterogeneous data collected by IoT networks into homogeneous data, which serves to increase the efficiency of the algorithms used to monitor and control IoT networks.
- It has used formal mechanisms to solve the problem of optimising energy consumption in smart buildings. In this sense, the proposed architecture and the designed algorithms have increased the efficiency of the techniques and technologies used for monitoring and control in IoT networks in smart buildings.
- It has demonstrate the remarkable performance of the proposed architecture and designed algorithms by means of their implementation and assessment in realistic scenarios. This implementation has been tested for monitoring and temperature control in smart buildings by simulation in different scenarios and by experimentation in real case studies. The derived results have pointeed out the noticeable improvement in energy efficiency in comparison with state-of-the-art techniques. Different simulations and real case studies to prove their efficiency. The results obtained in real application environments have been empirically evaluated.

In conclusion, it should be noted that this Doctoral Thesis has achieved the initial objectives set for it at the outset:

- New methods have been proposed to optimize information routing in IoT architectures and to transform heterogeneous data into homogeneous data collected by IoT networks.
- New techniques have been developed to increase the reliability of the data collected by the dynamic IoT network has been increased and to optimize the preventive maintenance of the dynamic IoT networks has been optimized.

- A new control algorithm has been designed to improve fault tolerance in dynamic IoT networks and to increase the robustness of the IoT network.
- New algorithms have been derived to optimize the monitoring and control tasks in dynamic IoT networks, and to improve energy consumption in smart buildings with dynamic IoT networks.
- Realistic simulation and empirical experimentation have been carried out to verify the efficiency of the proposed techniques and to demonstrate the reduction in energy consumption in comparison with state-of-the-art techniques.

The main objective of this doctoral thesis is to model new modular techniques and algorithms as an effective mechanism for the optimization of monitoring and control of the temperature in IoT dynamical networks and for the reduction of the energy consumption in smart buildings. As explained above, this initial objective has been successfully achieved.

7.3 State-of-the art contributions

The research work presented in this Doctoral thesis provides some new contributions in the fields of monitoring and control in IoT network in smart buildings in order to optimize the energy consumption:

- From the engineering point of view, a methodology oriented to the development of algorithms has been used to cover the analysis, design and development of these algorithms in an efficient way for monitoring and control in IoT networks in smart buildings.
- From a development point of view, the validity of this type of model could be verified in a real environment, which consists of the optimization of monitoring and temperature control in smart buildings with dynamic IoT networks.
- The results obtained have been analysed and studied in order to demonstrate that the proposed model are a viable solution for optimising energy consumption in smart buildings with dynamic IoT networks.
- Significant work has been done to obtain feedback from different researchers and research groups in related areas. The aim has been to strengthen this research through the mutual exchange of ideas and knowledge. Special interest has been placed on disseminating our experiences and advances in this research, from its

initial stages to its final form, through publications, attendance at congresses, international stays and the organization of workshops.

In addition, the scientific publications and other works derived from the above-mentioned contributions are listed in the Appendix A.

7.4 Future work

The research presented in this doctoral thesis validates the model proposed for the optimization of energy consumption in IoT networks in smart buildings. This result is only a starting point at a time when it is foreseeable a clear expansion of smart cities and in particular of smart buildings and IoT technology. In this sense, the main research line of the future work will be the inclusion of new mathematical techniques to optimize the functioning of algorithms, as well as the use of population dynamics to study the behavior of IoT devices and try to predict their indices of accuracy. This will allow to improve the preventive maintenance of IoT networks, which is a great advance in the optimization of energy consumption in smart buildings. This is so, because if the IoT network is very efficient in monitoring and controlling the temperature of a smart building, only the strictly necessary resources will be used and no energy will be wasted. Finally, the last line of research that is opened, for the moment, related to this research is the use of algorithms, techniques and technologies described in this research work with other environmental variables, such as humidity, in smart buildings.

Based on the results obtained in this work, other possible lines of work can be specified in the following aspects:

- Testing and validation. It is necessary to carry out many more exhaustive tests in order to evaluate in detail the architecture and the algorithms proposed in terms of time, application of analysis and design, quality of the monitoring and control of smart buildings, etc. The results obtained will allow the development of more refined and robust algorithms and systems.
- Solving new practical problems. In order to carry out a more exhaustive verification of the validity of the proposed algorithms, it is necessary to apply it to new practical problems and with new variables apart from the temperature. In this way, it could be verified if it adapts in an adequate way to the resolution of new problems.
- Incorporation of new techniques. In order to improve and optimize current algorithms it is necessary to use new mathematical techniques to design and

implement new modular algorithms in dynamic IoT networks in smart buildings architectures. Therefore, with more efficient algorithms, the task of monitoring and controlling smart buildings would be more effective and therefore energy would be saved in smart buildings.

Apéndice A

Publication and related works

VNiVERSiDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Apéndice A

Publication and related works

A.1 Introduction

Below is a list of some of the most relevant publications related to this work, which have been published since enrolment in the first year of doctoral studies, initially in international journals and later as book chapters. Next, we present the R&D projects in which we have participated and finally, we present the intellectual properties obtained as a consequence of aspects of the work presented.

A.1.1 Papers in international journals

- **Casado-Vara, R.**, Chamoso, P., De la Prieta, F., Prieto, J., & Corchado, J. M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion. **JCR 2017: 6.639, Q1. 4/101 (Computer science, theory & Methords)**.

- **Casado-Vara, R.**, Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access. **JCR 2017: 3.557, Q1. 24/148 (Computer science, Information systems)**.

- **Casado-Vara, R.**, Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102. **JCR 2017: 3.856, Q1. 4/252 (Mathematics, Applied)**.

- I. Sittón-Candanedo, R.S. Alonso, J.M. Corchado, S. Rodríguez & **R. Casado-Vara**. A review of edge computing reference architectures and a new global edge proposal, Future Generation Computer Systems (2019). **JCR 2017: 4.639, Q1. 7/101 (Computer science, theory & Methords)**.

- **Casado-Vara, R.**, Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for iot networks in smart buildings. Energies, 11(12), 3430. **JCR 2017: 2.676, Q2. 48/95 (Energy & fuels)**.

- **Casado-Vara, R.**, & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386. **JCR 2017: 1.426, Q3. 76/132 (Computer science, Artificial intelligence)**.

A.1.2 Book chapters

- **Casado-Vara, R.**, González-Briones, A., Prieto, J., & Corchado, J. M. (2018, June). Smart contract for monitoring and control of logistics activities: Pharmaceutical utilities case study. In The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications (pp. 509-517). Springer, Cham.

- **Casado-Vara, R.**, Prieto, J., De la Prieta, F., & Corchado, J. M. (2018). How blockchain improves the supply chain: Case study alimentary supply chain. Procedia computer science, 134, 393-398.

- **Casado-Vara, R.**, Prieto, J., & Corchado, J. M. (2018, June). How blockchain could improve fraud detection in power distribution grid. In The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications (pp. 67-76). Springer, Cham.

- González-Briones, A., Rivas, A., Chamoso, P., **Casado-Vara, R.**, & Corchado, J. M. (2018, June). Case-based reasoning and agent based job offer recommender system. In The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications (pp. 21-33). Springer, Cham.

- González-Briones, A., Valdeolmillos, D., **Casado-Vara, R.**, Chamoso, P., Coria, J. A. G., Herrera-Viedma, E., & Corchado, J. M. (2018, June). Garbmas: Simulation of the application of gamification techniques to increase the amount of recycled waste through a multi-agent system. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 332-343). Springer, Cham.

- **Casado-Vara, R.**, Chamoso, P., Coria, J. A. G., Herrera-Viedma, E., & Corchado, J. M. (2018, July). Garbmas: Simulation of the application of gamification techniques to increase the amount of recycled waste through a multi-agent system. In Distributed Computing and Artificial Intelligence, 15th International Conference (Vol. 800, p. 332). Springer.

- **Casado-Vara, R.**, de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (pp. 19-24). ACM.

- **Casado-Vara, R.**. (2018, June). Blockchain-Based Distributed Cooperative Control Algorithm for WSN Monitoring. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 414-417). Springer, Cham.

- **Casado-Vara, R.** (2018, June). Stochastic Approach for Prediction of WSN Accuracy Degradation with Blockchain Technology. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 422-425). Springer, Cham.

- **Casado-Vara, R.** (2018, June). New Approach to Power System Grid Security with a Blockchain-Based Model. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 418-421). Springer, Cham.

- González-Briones, A., Rivas, A., Chamoso, P., **Casado-Vara, R.**, & Corchado, J. M. (2018, June). Case-based reasoning and agent based job offer recommender system. In The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications (pp. 21-33). Springer, Cham.

- Cordero-Gutiérrez, R., Chamoso, P., Briones, A. G., Rivas, A., **Casado-Vara, R.**, & Corchado, J. M. (2018, June). The Right to Honour on Social Networks: Detection and Classifications of Users. In The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications (pp. 90-99). Springer, Cham.

A.1.3 Project participation

- **Title**: IOTEC: Desarrollo de Capacidades Tecnológicas en torno a la Aplicación Industrial de Internet de las Cosas (IoT).

  Founder: Fondo Europeo de Desarrollo Regional (FEDER) (Interreg España-Portugal (PocTep))

- **Title**: CITIES: Ciudades Inteligentes Totalmente Integrales, Eficientes y Sostenibles.

  Founder: CYTED(Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo)

- **Title**: HERMES: Hybrid Enhanced Regenerative Medicine Systems.

  Founder: European Commission(H2020-FETPROACT-2018-2020).

- **Title**: Dream-Go: Enabling Demand Response for short and real-time Efficient And Market Based smart Grid Operation - An intelligent and real-time simulation approach.

  Founder: Comisión Europea(Horizon 2020. MSCA-RISE-2014: Marie Skłodowska-Curie Research and Innovation Staff Exchange (RISE))

- **Title**: SURF: Arquitectura autoorganizativa de sensores y biometría para el control dinámico de vehículos en ciudades inteligentes

  Founder: Ministerio de Ciencia e Innovación. Proyectos de Investigación Fundamental No Orientada

- **Title**: CHROMOSOME: Change and Analysis of Consumer Behavior at Smart Homes via Social Machine

  Founder: Fundación Salamanca Ciudad de Cultura y Saberes

Bibliography

Abreu, D. P., Velasquez, K., Curado, M., and Monteiro, E. (2017). A resilient internet of things architecture for smart cities. *Annals of Telecommunications*, 72(1-2):19–30.

Afsar, M. (2015). Energy-Efficient Coalition Formation in Sensor Networks: a Game-Theoretic Approach.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM.

Alaayed, I., El Bahja, H., and Vega, P. (2013). A sliding mode based on fuzzy logic control for photovoltaic power system using dc-dc boost converter. In *3rd International Conference on Systems and Control*, pages 320–325. IEEE.

Alenezi, A., Almeraj, Z., and Manuel, P. (2018). Challenges of iot based smart-government development. In *Green Technologies Conference (GreenTech), 2018*, pages 155–160. IEEE.

Alonso, R. S., Tapia, D. I., Bajo, J., García, Ó., De Paz, J. F., and Corchado, J. M. (2013). Implementing a hardware-embedded reactive agents platform based on a service-oriented architecture over heterogeneous wireless sensor networks. *Ad Hoc Networks*, 11(1):151–166.

Alwi, H. and Edwards, C. (2008). Fault tolerant control using sliding modes with on-line control allocation. *Automatica*, 44(7):1859–1866.

Amato, F., Ariola, M., and Cosentino, C. (2006). Finite-time stabilization via dynamic output feedback. *Automatica*, 42(2):337–342.

Amato, F., Ariola, M., and Cosentino, C. (2010). Finite-time control of discrete-time linear systems: analysis and design conditions. *Automatica*, 46(5):919–924.

Anthopoulos, L., Janssen, M., and Weerakkody, V. (2019). A unified smart city model (uscm) for smart city conceptualization and benchmarking. In *Smart Cities and Smart Spaces: Concepts, Methodologies, Tools, and Applications*, pages 247–264. IGI Global.

Antonopoulos, A. M. (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.".

Åström, K. J., Hägglund, T., and Astrom, K. J. (2006). *Advanced PID control*, volume 461. ISA-The Instrumentation, Systems, and Automation Society Research Triangle ....

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

Axon, L. (2015). Privacy-awareness in blockchain-based pki.

Babar, M., Arif, F., and Irfan, M. (2019). Internet of things–based smart city environments using big data analytics: A survey. In *Recent Trends and Advances in Wireless and IoT-enabled Networks*, pages 129–138. Springer.

Bajo, J., De Paz, J., Villarrubia, G., and Corchado, J. (2015). Self-organizing architecture for information fusion in distributed sensor networks. *International Journal of Distributed Sensor Networks*, 2015.

Basin, M., Li, L., Krueger, M., and Ding, S. X. (2015). Finite-time-convergent fault-tolerant control for dynamical systems and its experimental verification for dts200 three-tank system. *IET Control Theory & Applications*, 9(11):1670–1675.

Beasley, J. and Christofides, N. (1997). Vehicle routing with a sparse feasibility graph. *European Journal of Operational Research*, 98(3):499–511.

Becerra-Bonache, L. and López, M. D. J. (2014). Linguistic models at the crossroads of agents, learning and formal languages. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 3(4):67–87.

Bishop, C. M. (2012). Pattern recognition and machine learning (information science and statistics), 2006. *Springer*, 60(1).

Blanchard, B. S., Fabrycky, W. J., and Fabrycky, W. J. (1990). *Systems engineering and analysis*, volume 4. Prentice Hall Englewood Cliffs, NJ.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271.

Bollen, J., Van de Sompel, H., Hagberg, A., and Chute, R. (2009). A principal component analysis of 39 scientific impact measures. *PloS one*, 4(6):e6022.

Boškovic, J. D. and Mehra, R. K. (2002). Multiple-model adaptive flight control scheme for accommodation of actuator failures. *Journal of Guidance, Control, and Dynamics*, 25(4):712–724.

Botta, A., De Donato, W., Persico, V., and Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56:684–700.

Bowman, C. L. and Steinberg, A. N. (2008). Revisions to the jdl data fusion model. In *Handbook of Multisensor Data Fusion*, pages 65–88. CRC Press.

Bremer, J. and Lehnhoff, S. (2017). Decentralized coalition formation with agent-based combinatorial heuristics. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(3):29–44.

Bui, K.-H. N., Jung, J. J., and Camacho, D. (2018). Consensual negotiation-based decision making for connected appliances in smart home management systems. *Sensors (Basel, Switzerland)*, 18(7).

Calvo-Rolle, J. L., Quintian-Pardo, H., Corchado, E., del Carmen Meizoso-López, M., and García, R. F. (2015). Simplified method based on an intelligent model to obtain the extinction angle of the current for a single-phase half wave controlled rectifier with resistive and inductive load. *Journal of Applied Logic*, 13(1):37–47.

Cao, Y., Qi, H., Zhou, W., Kato, J., Li, K., Liu, X., and Gui, J. (2018). Binary hashing for approximate nearest neighbor search on big data: A survey. *IEEE Access*, 6:2039–2054.

Capellari, G., Chatzi, E., and Mariani, S. (2018). Cost–benefit optimization of structural health monitoring sensor networks. *Sensors (Basel, Switzerland)*, 18(7).

Cardoso, R. C. and Bordini, R. H. (2017). A multi-agent extension of a hierarchical task network planning formalism. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):5–17.

Caronni, G. (2000). Walking the web of trust. In *Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*, pages 153–158.

Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto, J., and Corchado, J. M. (2019a). Non-linear adaptive closed-loop control system for improved efficiency in iot-blockchain management. *Information Fusion*, 49:227–239.

Casado-Vara, R. and Corchado, J. M. (2018). Blockchain for democratic voting: how blockchain could cast off voter fraud. *Orient. J. Comp. Sci. and Technol*, 11(1).

Casado-Vara, R., González-Briones, A., Prieto, J., and Corchado, J. M. (2018a). Smart contract for monitoring and control of logistics activities: Pharmaceutical utilities case

study. In *The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications*, pages 509–517. Springer.

Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., and Corchado, J. M. (2019b). Distributed continuous-time fault estimation control for multiple devices in iot networks. *IEEE Access*.

Casado-Vara, R., Prieto-Castrillo, F., and Corchado, J. M. (2018b). A game theory approach for cooperative control to improve data quality and false data detection in wsn. *International Journal of Robust Nonlinear Control*.

Casado-Vara, R., Vale, Z., Prieto, J., and Corchado, J. (2018c). Fault-tolerant temperature control algorithm for iot networks in smart buildings. *Energies*, 11(12):3430.

Casteleiro-Roca, J. L., Pérez, J. A. M., Piñón-Pazos, A. J., Calvo-Rolle, J. L., and Corchado, E. (2015). Modeling the electromyogram (emg) of patients undergoing anesthesia during surgery. In *10th international conference on soft computing models in industrial and environmental applications*, pages 273–283. Springer.

Chamoso, P., González-Briones, A., Rodríguez, S., and Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. *Wireless Communications and Mobile Computing*, 2018.

Chekired, D. A., Khoukhi, L., and Mouftah, H. T. (2018). Industrial iot data scheduling based on hierarchical fog computing: A key for enabling smart factory. *IEEE Transactions on Industrial Informatics*.

Chen, X., Huo, H., Huan, J., and Vitter, J. S. (2016). Msq-index: A succinct index for fast graph similarity search. *arXiv preprint arXiv:1612.09155*.

Choi, C.-S., Baccelli, F., and de Veciana, G. (2018). Densification leveraging mobility: An iot architecture based on mesh networking and vehicles. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 71–80. ACM.

Cirani, S., Davoli, L., Ferrari, G., Léone, R., Medagliani, P., Picone, M., and Veltri, L. (2014). A scalable and self-configuring architecture for service discovery in the internet of things. *IEEE Internet of Things Journal*, 1(5):508–521.

Coley, D., Kershaw, T., and Eames, M. (2012). A comparison of structural and behavioural adaptations to future proofing buildings against higher temperatures. *Building and Environment*, 55:159–166.

Conoscenti, M., Vetro, A., and De Martin, J. C. (2016). Blockchain for the internet of things: A systematic literature review. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–6. IEEE.

Conoscenti, M., Vetrò, A., and Martin, J. C. D. (2017). Peer to peer for privacy and decentralization in the internet of things. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 288–290.

Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298.

Corchado, J. M., Bajo, J., Tapia, D. I., and Abraham, A. (2010). Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 14(2):234–40.

Costa, I. G., de Carvalho, F. d. A., and de Souto, M. C. (2004). Comparative analysis of clustering methods for gene expression time course data. *Genetics and Molecular Biology*, 27(4):623–631.

de Paz, J. F., Tapia, D. I., Alonso, R. S., Pinzón, C. I., Bajo, J., and Corchado, J. M. (2013). Mitigation of the ground reflection effect in real-time locating systems based on wireless sensor networks by using artificial neural networks. *Knowledge and Information Systems*, 34(1):193–217.

de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B., and Schliep, A. (2008). Clustering cancer gene expression data: a comparative study. *BMC bioinformatics*, 9(1):497.

Dong, H., Wang, Z., Ding, S. X., and Gao, H. (2016). On h-infinity estimation of randomly occurring faults for a class of nonlinear time-varying systems with fading channels. *IEEE Transactions on Automatic Control*, 61(2):479–484.

Dong, J. and Yang, G.-H. (2015). Reliable state feedback control of t–s fuzzy systems with sensor faults. *IEEE Transactions on Fuzzy Systems*, 23(2):421–433.

Efficiency, E. (2009). Buildings energy data book. *US Department of Energy. http://buildingsdatabook. eere. energy. gov/. John Dieckmann is a director and Alissa Cooperman is a technologist in the Mechanical Systems Group of TIAX, Cambridge, Mass. James Brodrick, Ph. D., is a project manager with the Building Technologies Program, US Department of Energy, Washington, DC.*

El Bahja, H., Vega, P., and Revollar, S. (2014). Economic optimization based on nonlinear parametric gpc for a wastewater treatment plant. In *53rd IEEE Conference on Decision and Control*, pages 3815–3820. IEEE.

Fairchild, A. (2019). Twenty-first-century smart facilities management: Ambient networking in intelligent office buildings. In *Guide to Ambient Intelligence in the IoT Environment*, pages 271–289. Springer.

Fattahi, M. and Afshar, A. (2018). Controller-based observer design for distributed consensus of multi-agent systems with fault and delay. *Journal of Control and Decision*, pages 1–19.

Fomundam, S. and Herrmann, J. W. (2007). A survey of queuing theory applications in healthcare. Technical report.

Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588.

Friese, I., Heuer, J., and Kong, N. (2014). Challenges from the identities of things: Introduction of the identities of things discussion group within kantara initiative. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 1–4. IEEE.

Fromknecht, C., Velicanu, D., and Yakoubov, S. (2014). Certcoin: A namecoin based decentralized authentication system. In *Technical Report*. Massachusetts Institute of Technology, MA, USA. 6.857 Class Project.

Gao, Z., Cecati, C., and Ding, S. X. (2015a). A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3757–3767.

Gao, Z., Ding, S. X., and Cecati, C. (2015b). Real-time fault diagnosis and fault-tolerant control. *IEEE Transactions on industrial Electronics*, 62(6):3752–3756.

Garcia, R. F., Rolle, J. L. C., Castelo, J. P., and Gomez, M. R. (2014). On the monitoring task of solar thermal fluid transfer systems using nn based models and rule based techniques. *Engineering Applications of Artificial Intelligence*, 27:129–136.

Ge, X., Yang, F., and Han, Q.-L. (2017). Distributed networked control systems: A brief overview. *Information Sciences*, 380:117–131.

Giambene, G. (2005). *Queuing theory and telecommunications*. Springer.

Gnedenko, B. and Kovalenko, I. (1989). *Introduction to Queuing Theory. Mathematical Modeling*. Birkhaeuser Boston, Boston.

Golder, S. A. and Macy, M. W. (2011). Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881.

Guha, S., Rastogi, R., and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. In *ACM Sigmod Record*, volume 27, pages 73–84. ACM.

Gungor, V. C., Hancke, G. P., and Member, S. (2009). Industrial Wireless Sensor Networks : Challenges , Design Principles , and Technical Approaches. *4258 Ieee Transactions on Industrial Electronics*, 56(10):4258–4265.

Gupta, P., Mokal, T. P., Shah, D., and Satyanarayana, K. (2018). Event-driven soa-based iot architecture. In *International Conference on Intelligent Computing and Applications*, pages 247–258. Springer.

Haibo, L. and Fang, Z. (2015). Design and Implementation of a Wireless Sensor Network Testbed. *Ncc 2015*, 49(November):792–797.

Han, Z., Niyato, D., Saad, W., Başar, T., & Hjørungnes, A. (2011). *Game theory in wireless and communication networks: Theory, models, and applications.* Cambridge University Press.

Herbert, J. and Litchfield, A. (2015). A novel method for decentralised peer-to-peer software license validation using cryptocurrency blockchain technology. In *Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015)*, volume 27, page 30.

Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162.

Hui, Q., Haddad, W. M., and Bhat, S. P. (2008). Finite-time semistability and consensus for nonlinear dynamical networks.

Jain, A. K., Topchy, A., Law, M. H., and Buhmann, J. M. (2004). Landscape of clustering algorithms. In *null*, pages 260–263. IEEE.

Jin, X. (2016). Adaptive fault tolerant control for a class of input and state constrained mimo nonlinear systems. *International Journal of Robust and Nonlinear Control*, 26(2):286–302.

Kaufmann, A. (1973). *Introduction à la théorie des sous-ensembles flous à l'usage des ingénieurs: Éléments théoriques de base*, volume 1. Masson.

Khan, M. A. and Salah, K. (2018). Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411.

Khan, M. W. and Zeeshan, M. (2019). Qos-based dynamic channel selection algorithm for cognitive radio based smart grid communication network. *Ad Hoc Networks*, 87:61–75.

Khoo, S., Xie, L., Zhao, S., and Man, Z. (2014). Multi-surface sliding control for fast finite-time leader–follower consensus with high order siso uncertain nonlinear agents. *International Journal of Robust and Nonlinear Control*, 24(16):2388–2404.

Kinnunen, T., Sidoroff, I., Tuononen, M., and Fränti, P. (2011). Comparison of clustering methods: A case study of text-independent speaker modeling. *Pattern Recognition Letters*, 32(13):1604–1617.

Kleinrock, L. (1976). *Queueing systems, volume 2: Computer applications*, volume 66. wiley New York.

Korobov, V., Pavlichkov, S., and Schmidt, W. (2013). Global positional synthesis and stabilization in finite time of mimo generalized triangular systems by means of the controllability function method. *Journal of Mathematical Sciences*, 189(5):795–804.

Kottapalli, V. A., Kiremidjian, A. S., Lynch, J. P., Carryer, E., Kenny, T. W., Law, K. H., and Lei, Y. (2003). Two-tiered wireless sensor network architecture for structural health monitoring. *Proceedings of SPIE*, 5057:8–19.

Kraijak, S. and Tuwanut, P. (2015). A survey on iot architectures, protocols, applications, security, privacy, real-world implementation and future trends.

Lapillonne, B., Sebi, C., Pollier, K., and Mairet, N. (2012). Energy efficiency trends in buildings in the eu—lessons from the odyssee mure project. *ADEME. Retrieved December*, 30:2014.

Leu, J.-S., Chen, C.-F., and Hsu, K.-C. (2014). Improving heterogeneous soa-based iot message stability by shortest processing time scheduling. *IEEE Transactions on Services Computing*, 7(4):575–585.

Li, J., Dong, H., Han, F., Hou, N., and Li, X. (2017a). Filter design, fault estimation and reliable control for networked time-varying systems: a survey. *Systems Science & Control Engineering*, 5(1):331–341.

Li, L., Ding, S. X., Qiu, J., Peng, K., and Yang, Y. (2017b). An optimal fault detection approach for piecewise affine systems via diagnostic observers. *Automatica*, 85:256–263.

Li, X.-J. and Yang, G.-H. (2012). Robust adaptive fault-tolerant control for uncertain linear systems with actuator failures. *IET control theory & applications*, 6(10):1544–1551.

Li, Y., Tong, S., and Li, T. (2015). Composite adaptive fuzzy output feedback control design for uncertain nonlinear strict-feedback systems with input saturation. *IEEE Transactions on Cybernetics*, 45(10):2299–2308.

Li, Y.-X. and Yang, G.-H. (2018). Event-triggered adaptive backstepping control for parametric strict-feedback nonlinear systems. *International Journal of Robust and Nonlinear Control*, 28(3):976–1000.

Liu, K., Seuret, A., Fridman, E., and Xia, Y. (2018). Improved stability conditions for discrete-time systems under dynamic network protocols. *International Journal of Robust and Nonlinear Control*, pages 1–21.

Lorincz, K., Malan, D. J., Fulford-jones, T. R. F., Nawoj, A., Clavel, A., Shanyder, V., Mainland, G., Welsh, M., and Moulton, S. (2004). Sensor Networks for Emergency Response: Challenges and Opportunities. *Pervasive Computing*, 3(4):16–23.

Machón González, I. J., López García, H., and Calvo Rolle, J. L. (2011). Neuro-robust controller for non-linear systems (controlador neurorobusto para sistemas no lineales). *Dyna*.

Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L., and Vert, J.-P. (2005). Graph kernels for molecular structure- activity relationship analysis with support vector machines. *Journal of chemical information and modeling*, 45(4):939–951.

Mahmoud, R., Yousuf, T., Aloul, F., and Zualkernan, I. (2015). Internet of things (iot) security: Current status, challenges and prospective measures. In *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for*, pages 336–341. IEEE.

Mailund, T. (2018). Continuous-time markov chains. In *Domain-Specific Languages in R*, pages 167–182. Springer.

Manogaran, G., Varatharajan, R., Lopez, D., Kumar, P. M., Sundarasekar, R., and Thota, C. (2018). A new architecture of internet of things and big data ecosystem for secured smart healthcare monitoring and alerting system. *Future Generation Computer Systems*, 82:375–387.

Manuel Vilar-Martinez, X., Aurelio Montero-Sousa, J., Luis Calvo-Rolle, J., and Luis Casteleiro-Roca, J. (2014). Expert system development to assist on the verification of" tacan" system performance. *Dyna*, 89(1):112–121.

Marín, P. A. R., Duque, N., and Ovalle, D. (2015). Multi-agent system for knowledge-based recommendation of learning objects. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 4(1):80–89.

Marin, R. M., Aguirre, N. F., and Daza, E. E. (2008). Graph theoretical similarity approach to compare molecular electrostatic potentials. *Journal of chemical information and modeling*, 48(1):109–118.

Marques, P., Manfroi, D., Deitos, E., Cegoni, J., Castilhos, R., Rochol, J., Pignaton, E., and Kunst, R. (2019). An iot-based smart cities infrastructure architecture applied to a waste management scenario. *Ad Hoc Networks*, 87:200–208.

Matias, J., Garay, J., Toledo, N., Unzilla, J., and Jacob, E. (2015). Toward an sdn-enabled nfv architecture. *IEEE Communications Magazine*, 53(4):187–193.

Medhi, J. (2002). *Stochastic models in queueing theory*. Elsevier.

Michael, A. J. and Mohammad, H. M. (2005). Pid control new identification and design methods. *London: Spring-Verlag London Limited*.

Miller, D. (2018). Blockchain and the internet of things in the industrial sector. *IT Professional*, 20(3):15–18.

Minoli, D., Sohraby, K., and Occhiogrosso, B. (2017). Iot considerations, requirements, and architectures for smart buildings—energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, 4(1):269–283.

Mishra, R., Jha, V., Tripathi, R. K., and Sharma, A. K. (2017). Energy Efficient Approach in Wireless Sensor Networks Using Game Theoretic Approach and Ant Colony Optimization. *Wireless Personal Communications*, 95(3):3333–3355.

Mo, Y., Garone, E., Casavola, A., and Sinopoli, B. (2010). False data injection attacks against state estimation in wireless sensor networks. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5967–5972. IEEE.

Monteriù, A., Prist, M. R., Frontoni, E., Longhi, S., Pietroni, F., Casaccia, S., Scalise, L., Cenci, A., Romeo, L., Berta, R., et al. (2018). A smart sensing architecture for domestic monitoring: Methodological approach and experimental validation. *Sensors (Basel, Switzerland)*, 18(7).

Moodie, J., Teräs, J., and Randall, L. (2018). Building effective transnational partnerships: The case of smart lighting.

Moura, J. and Hutchison, D. (2017). Survey of Game Theory and Future Trends with Application to Emerging Wireless Data Communication Networks. *IEEE Communications Surveys & Tutorials*, pages 1–50.

Muthanna, A., Ateya, A. A., Khakimov, A., Gudkova, I., Abuarqoub, A., Samouylov, K., and Koucheryavy, A. (2019). Secure iot network structure based on distributed fog computing, with sdn/blockchain.

Nakamoto, S. (2008). A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`. 2018-08-30.

Novo, O. (2018). Blockchain meets iot: an architecture for scalable access management in iot. *IEEE Internet of Things Journal*.

Panarello, A., Tapas, N., Merlino, G., Longo, F., and Puliafito, A. (2018). Blockchain and iot integration: A systematic survey. *Sensors*, 18(8):2575.

Patel, K. K., Patel, S. M., et al. (2016). Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6(5).

Patel, S. V. and Pandey, K. (2010). Design of SOA Based Framework for Collaborative Cloud Computing in Wireless Sensor Networks. *Int. J. Grid High Perform. Comput.*, 2(3):60–73.

Perruquetti, W., Floquet, T., and Moulay, E. (2008). Finite-time observers: application to secure communication. *IEEE Transactions on Automatic Control*, 53(1):356–360.

Petrolo, R., Mitton, N., Soldatos, J., Hauswirth, M., and Schiele, G. (2014). Integrating wireless sensor networks within a city cloud. In *Proceedings of SWANSITY-International IEEE SECON Workshop on Self-Organizing Wireless Access Networks for Smart City*.

Pilloni, V., Floris, A., Meloni, A., and Atzori, L. (2018). Smart home energy management including renewable sources: A qoe-driven approach. *IEEE Transactions on Smart Grid*, 9(3):2006–2018.

Pipino, L. L., Lee, Y. W., and Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM*, 45(4):211–218.

Polyakov, A., Efimov, D., and Perruquetti, W. (2016). Robust stabilization of mimo systems in finite/fixed time. *International Journal of Robust and Nonlinear Control*, 26(1):69–90.

Pop, C., Cioara, T., Antal, M., Anghel, I., Salomie, I., and Bertoncini, M. (2018). Blockchain based decentralized management of demand response programs in smart energy grids. *Sensors*, 18(1):162.

Qin, Z., Denker, G., Giannelli, C., Bellavista, P., and Venkatasubramanian, N. (2014). A software defined networking architecture for the internet-of-things. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–9. IEEE.

Qiu, A., Gu, J., Wen, C., and Zhang, J. (2018). Self-triggered fault estimation and fault tolerant control for networked control systems. *Neurocomputing*, 272:629–637.

Qiu, J., Gao, H., and Chow, M.-Y. (2016). Networked control and industrial applications [special section introduction]. *IEEE Transactions on Industrial Electronics*, 63(2):1203–1206.

Quintian Pardo, H., Calvo Rolle, J. L., and Fontenla Romero, O. (2012). Application of a low cost commercial robot in tasks of tracking of objects. *Dyna*, 79(175):24–33.

Rahimi, H., Zibaeenejad, A., and Safavi, A. A. (2018). A novel iot architecture based on 5g-iot and next generation technologies. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 81–88. IEEE.

Rao, C. R., Rao, C. R., Statistiker, M., Rao, C. R., and Rao, C. R. (1973). *Linear statistical inference and its applications*, volume 2. Wiley New York.

Ray, P. P. (2018). A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3):291–319.

Ray, S., Jin, Y., and Raychowdhury, A. (2016). The changing computing paradigm with internet of things: A tutorial introduction. *IEEE Design & Test*, 33(2):76–96.

Reason, P. and Bradbury, H. (2001). *Handbook of action research: Participative inquiry and practice*. Sage.

Ren, W., Wang, C., and Lu, Y. (2017). Fault estimation for time-varying markovian jump systems with randomly occurring nonlinearities and time delays. *Journal of the Franklin Institute*, 354(3):1388–1402.

Rodríguez, S., De Paz, J. F., Villarrubia, G., Zato, C., Bajo, J., and Corchado, J. M. (2015). Multi-agent information fusion system to manage data from a WSN in a residential home. *Information Fusion*, 23:43–57.

Saghezchi, F. B., Radwan, A., and Rodriguez, J. (2017). Energy-aware relay selection in cooperative wireless networks: An assignment game approach. *Ad Hoc Networks*, 56:96–108.

Samie, F., Bauer, L., and Henkel, J. (2019). From cloud down to things: An overview of machine learning in internet of things. *IEEE Internet of Things Journal*.

Sampathkumar, E. (2006). Generalized graph structures. *Bull. Kerala Math. Assoc*, 3(2):65–123.

Samuel, R. T. and Cao, Y. (2016). Nonlinear process fault detection and identification using kernel pca and kernel density estimation. *Systems Science & Control Engineering*, 4(1):165–174.

Sánchez Fernández, A., Francisco Sutil, M., de la Fuente Aparicio, M. J., and Vega Cruz, P. (2016). Control predictivo no lineal tolerante a fallos en una planta de tratamiento de aguas residuales.

Sarwesh, P., Shet, N. S. V., and Chandrasekaran, K. (2019). Envisioned network architectures for iot applications. In *Cyber-Physical Systems: Architecture, Security and Application*, pages 3–17. Springer.

Schmidt, C. (2002). *Game Theory and Economic Analysis*.

Shahnazari, H. and Mhaskar, P. (2018). Actuator and sensor fault detection and isolation for nonlinear systems subject to uncertainty. *International Journal of Robust and Nonlinear Control*, 28(6):1996–2013.

Sharma, P. K., Park, J. H., Jeong, Y.-S., and Park, J. H. (2018). Shsec: sdn based secure smart home network architecture for internet of things. *Mobile Networks and Applications*, pages 1–12.

Shashwat, A., Kumar, D., and Chanana, L. (2018). Message level security enhancement for service oriented architecture. In *2018 4th International Conference on Computational Intelligence & Communication Technology (CICT)*, pages 1–6. IEEE.

Shi, H.-Y., Wang, W.-L., Kwok, N.-M., and Chen, S.-Y. (2012). Game theory for wireless sensor networks: a survey. *Sensors*, 12(7):9055–9097.

Sikder, A. K., Acar, A., Aksu, H., Uluagac, A. S., Akkaya, K., and Conti, M. (2018). Iot-enabled smart lighting systems for smart cities. In *Computing and Communication Workshop and Conference (CCWC), 2018 IEEE 8th Annual*, pages 639–645. IEEE.

Silva, B. N., Khan, M., and Han, K. (2018). Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. *IETE Technical Review*, 35(2):205–220.

Srivastava, R. (2018). Exploration of in-memory computing for big data analytics using queuing theory. In *Proceedings of the 2nd International Conference on High Performance Compilation, Computing and Communications*, pages 11–16. ACM.

Su, I.-J., Tsai, C.-C., and Sung, W.-T. (2012). Area temperature system monitoring and computing based on adaptive fuzzy logic in wireless sensor networks. *Applied Soft Computing*, 12(5):1532–1541.

Sun, W., Zeng, Z., Luo, C., and Nguang, S.-K. (2018). Robust-gain codesign of networked control systems. *International Journal of Robust and Nonlinear Control*, pages 1–15.

Sung, S. W., Lee, J., and Lee, I.-B. (2009). *Process identification and PID control*. John Wiley & Sons.

Tan, C., Li, L., and Zhang, H. (2015). Stabilization of networked control systems with both network-induced delay and packet dropout. *Automatica*, 59:194–199.

Tao, G., Chen, S., Tang, X., and Joshi, S. M. (2013). *Adaptive control of systems with actuator failures*. Springer Science & Business Media.

Tao, M., Zuo, J., Liu, Z., Castiglione, A., and Palmieri, F. (2018). Multi-layer cloud architectural model and ontology-based security service framework for iot-based smart homes. *Future Generation Computer Systems*, 78:1040–1051.

Tapia, D. I., Abraham, A., Corchado, J. M., and Alonso, R. S. (2010a). Agents and ambient intelligence: Case studies. *Journal of Ambient Intelligence and Humanized Computing*, 1(2):85–93.

Tapia, D. I., Alonso, R. S., De la Prieta, F., Zato, C., Rodriguez, S., Corchado, E., Bajo, J., and Corchado, J. M. (2010b). SYLPH: An Ambient Intelligence based platform for integrating heterogeneous Wireless Sensor Networks. *International Conference on Fuzzy Systems (FUZZ)*, (February 2016):1–8.

Tapia, D. I., Alonso, R. S., Paz, J. F. D., and Corchado, J. M. (2009). Introducing a Distributed Architecture for Heterogeneous Wireless Sensor Networks. pages 116–123.

Tatnall, A. and Davey, B. (2019). Rise of the non-human actors: The internet of things. In *Analytical Frameworks, Applications, and Impacts of ICT and Actor-Network Theory*, pages 138–155. IGI Global.

Terroso-Saenz, F., González-Vidal, A., Ramallo-González, A. P., and Skarmeta, A. F. (2019). An open iot platform for the management and analysis of energy data. *Future Generation Computer Systems*, 92:1066–1079.

Tiburski, R. T., Amaral, L. A., De Matos, E., and Hessel, F. (2015). The importance of a standard securit y archit ecture for soa-based iot middleware. *IEEE Communications Magazine*, 53(12):20–26.

Tom, R. J., Sankaranarayanan, S., and Rodrigues, J. J. (2019). Smart energy management and demand reduction by consumers and utilities in an iot-fog based power distribution system. *IEEE Internet of Things Journal*.

Uddin, M., Mukherjee, S., Chang, H., and Lakshman, T. (2018). Sdn-based multi-protocol edge switching for iot service automation. *IEEE Journal on Selected Areas in Communications*, 36(12):2775–2786.

Van Kranenburg, R. and Bassi, A. (2012). Iot challenges. *Communications in Mobile Computing*, 1(1):9.

Vellei, M., Natarajan, S., Biri, B., Padget, J., and Walker, I. (2016). The effect of real-time context-aware feedback on occupants' heating behaviour and thermal adaptation. *Energy and Buildings*, 123:179–191.

Vermesan, O. and Friess, P. (2013). *Internet of things: converging technologies for smart environments and integrated ecosystems*. River publishers.

Visioli, A. (2006). *Practical PID control*. Springer Science & Business Media.

Wang, H., Liu, K., Liu, X., Chen, B., and Lin, C. (2015). Neural-based adaptive output-feedback control for a class of nonstrict-feedback stochastic nonlinear systems. *IEEE transactions on cybernetics*, 45(9):1977–1987.

Wang, R. Y. (1998). A product perspective on total data quality management. *Communications of the ACM*, 41(2):58–65.

Wang, Y., Fan, Y., Bhatt, P., and Davatzikos, C. (2010). High-dimensional pattern regression using machine learning: from medical images to continuous clinical variables. *Neuroimage*, 50(4):1519–1535.

Watts, D. J., Dodds, P. S., and Newman, M. E. (2002). Identity and search in social networks. *science*, 296(5571):1302–1305.

Wiki, B. (2012). Irreversible transactions. `https://en.bitcoin.it/wiki/Irreversible_Transactions`. 2018-08-30.

Wiki, B. (2013). Hashcash. `https://en.bitcoin.it/wiki/Hashcash`. 2018-08-30.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wu, X., Hu, X., Yin, X., and Moura, S. J. (2018). Stochastic optimal energy management of smart home with pev energy storage. *IEEE Transactions on Smart Grid*, 9(3):2065–2075.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37.

Xie, Y. (2019). Intelligent sensor detection technology in lighting design and application. In *Proceedings of 2018 Chinese Intelligent Systems Conference*, pages 239–247. Springer.

Xu, K., Qu, Y., and Yang, K. (2016). A tutorial on the internet of things: from a heterogeneous network integration perspective. *IEEE Network*, 30(2):102–108.

Xu, Z., Yin, Y., Chen, X., and Wang, J. (2013). A Game-theory Based Clustering Approach for Wireless Sensor Networks. 27:58–66.

Xuan, J., Lu, J., Zhang, G., and Luo, X. (2015). Topic model for graph mining. *IEEE Trans. Cybernetics*, 45(12):2792–2803.

Yan, X. and Han, J. (2002). gspan: Graph-based substructure pattern mining. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 721–724. IEEE.

Yue, H., Guo, L., Li, R., Asaeda, H., and Fang, Y. (2014). Dataclouds: Enabling community-based data-centric services over the internet of things. *IEEE Internet of Things Journal*, 1(5):472–482.

Zadeh, L. A. (1971). Similarity relations and fuzzy orderings. *Information sciences*, 3(2):177–200.

Zhai, D., Zhang, R., Cai, L., Li, B., and Jiang, Y. (2018). Energy-efficient user scheduling and power allocation for noma based wireless networks with massive iot devices. *IEEE Internet of Things Journal*.

Zhang, J.-X. and Yang, G.-H. (2018). Fault-tolerant leader-follower formation control of marine surface vessels with unknown dynamics and actuator faults. *International Journal of Robust and Nonlinear Control*, 28(14):4188–4208.

Zhang, K., Jiang, B., and Shi, P. (2009). Fast fault estimation and accommodation for dynamical systems. *IET Control Theory & Applications*, 3(2):189–199.

Zhang, X. and Lin, Y. (2012). Adaptive output feedback tracking for a class of nonlinear systems. *Automatica*, 48(9):2372–2376.

Zhang, X. and Lin, Y. (2015). Adaptive output feedback control for a class of large-scale nonlinear time-delay systems. *Automatica*, 52:87–94.

Zhang, X., Parisini, T., and Polycarpou, M. M. (2004). Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach. *IEEE Transactions on automatic Control*, 49(8):1259–1274.

Zhang, X.-M., Han, Q.-L., and Yu, X. (2016). Survey on recent advances in networked control systems. *IEEE Transactions on Industrial Informatics*, 12(5):1740–1752.

Zhang, Y. and Wen, J. (2015). An iot electric business model based on the protocol of bitcoin. In *2015 18th International Conference on Intelligence in Next Generation Networks*, pages 184–191. IEEE.

Zhao, X., Xiao, C., Lin, X., and Wang, W. (2012). Efficient graph similarity joins with edit distance constraints. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 834–845. IEEE.

Zhou, J., Leppanen, T., Harjula, E., Ylianttila, M., Ojala, T., Yu, C., Jin, H., and Yang, L. T. (2013). Cloudthings: A common architecture for integrating the internet of things with cloud computing. In *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 651–657.

Zhou, Z., Zhang, H., Li, S., and Du, X. (2018). Hermes: a privacy-preserving approximate search framework for big data. *IEEE Access*, 6:20009–20020.

Zhou, J.; Mu, C. (2006). Density domination of QoS C ontrol with localized information in wireless sensor networks. *In Proceedings of 2006 6th International Conference on ITS Telecommunications. IEEE.*

Zou, H., Zhou, Y., Jiang, H., Chien, S.-C., Xie, L., and Spanos, C. J. (2018). Winlight: A wifi-based occupancy-driven lighting control system for smart building. *Energy and Buildings*, 158:924–938.

Zyskind, G., Nathan, O., and Pentland, A. (2015). Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471.*