République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA

Faculté des Sciences Exactes
Département d'Informatique

THÈSE

Présentée par

Sahar BOULKABOUL

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Cloud Computing

Thème

Data Processing in the Emerging Internet of Things

Soutenue le : 25/11/2021 Devant le Jury composé de :

Nom et Prénom	Grade		
Mr Nadjib BADACHE	Professeur	USTHB	Président
Mr Djamel DJENOURI	Directeur de Recherche	CERIST	Rapporteur
Mr Mahfoud BENCHAÏBA	Professeur	USTHB	Examinateur
Mr Djamel TANDJAOUI	Directeur de Recherche	CERIST	Examinateur
Mr Hachem SLIMANI	Professeur	Univ. de Bejaia	Examinateur
Mme Houda EL BOUHISSI	MCA	Univ. de Bejaia	Examinatrice

Année Universitaire : 2020/2021

Abstract

The emerging IoT (Internet of Things) is rapidly gaining field our modern society, aiming to improve the quality of life by connecting many smart devices, technologies, and applications, for the purpose of exchanging data over the Internet. Overall, the IoT would allow for the automation of everything around us. In the recent era of the Internet of Things, the dominant role of sensors and the Internet provides a solution to a wide variety of real life application domains including smart city, smart healthcare systems, smart building, smart transport and smart environment. Future applications will consist in the integration of IoT devices with other emerging technologies, such as edge computing, fog computing, cloud computing. Such technologies provide platforms that enable the development of flexible IoT services in the aim of recursively updating and adapting to the dynamic changes in the cyber physical system surrounding the IoT. IoT devices will generate huge volumes of data in a quick span of time and thus requires scalable solutions for dynamic and realtime processing of the generated data. Such solutions are expected to provide high level of accurate and reliable data for decision making. This calls for data/information fusion, which is an effective way for optimum utilization of large volumes of data from multiple sources.

We consider in this thesis IoT integration to edge,fog, and cloud computing, the efficiency of data processing and fusion in terms of data credibility, reliability, conflict, latency, and we propose several solutions herein. The first one is for efficient data processing in edge computing, which enables sophisticated services. The processing of data at the edge preserves the data privacy and the bandwidth when relaying data, while reducing the communication overheads. The second approach is an internet of things data management and control platform based on fog and cloud computing that allows heterogeneous resources, connectivity reliability and mobility, ensures security and contains services to fuse heterogeneity data in application framework architecture. The numerical analysis and simulation results show that the proposed solutions yields significant savings in energy consumption and delay reduction. The thesis considers also the state estimation in the medium level of data fusion. We propose an improved distributed particle filter algorithm to deal with target tracking in wireless sensors networks. It increase the estimation accuracy of the particle filter, enhance the efficiency of the particle sampling and improve the estimation performance. The principle contribution is a novel approach for calculating the similarity measure based on the distance between two or several belief functions. More precisely, we build probability density

functions induced by normal distribution representing continuous belief functions. Results of simulation and numerical analysis show the superiority of the proposed approach in terms of Root Mean Square Error and scalability. We have studied the problem of data fusion in decisional level. The reliability of devices in the network and the conflicts between devices are considered in our method by considering the information lifetime, the distance separating sensors and entities, reducing the computation and using combination rules based on the Basic Probability Assignment. This allows to represent uncertain information or to quantify the similarity between two bodies of evidence. We compared the proposed solution with some state-of-the-art data fusion methods, and using both benchmark data simulation and real dataset from a smart building testbed. Results show that our solution outperforms all the above mentioned methods in terms of reliability, accuracy and conflict management.

Dedication

This work is wholeheartedly dedicated to the memory of my father, Mohammed.

...To my mother Fedjria, who has been my source of inspiration and gave us stength, who consistently provided me with moral, spiritual and emotional support

...To my family: My husband Ali and my children Hani, Wail and Manil. To my sister Loubna, and my brothers: Adel, Imad, Issam, Mouloud, Mustapha, Hamza and Ibrahim. Through good times and bad, your kindness and extensive support have been ever present in this important time of my life, for which I am eternally grateful.

I dedicate this humble work to everyone I have had chance to know in this life.

...

Acknowledgements

First of all, I thank ALLAH for his help during all my life. I would never have completed this work and gotten this far without the help of ALLAH.

I would like to thank gratefully my supervisor, Prof. Djamel Djenouri, whose expertise was invaluable in formulating the research questions and methodology. Without any complaints he reviewed all my writings. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I am very grateful to express my sincere thanks and my deep gratitude to Prof. Nadjib Badache, for the opportunity to follow my studies at the doctoral school of bejaia. Many thanks are due to my colleagues at SenseLab, especially Elmouatezbillah Karbab, Cherif Zizoua, and Roufaida Laidi. In the lab, we work, as brothers and sisters, together to achieve the goals. Many thanks for their helps during my dissertation. I would like to thank my entire family. Special thanks to my mother, and husband who have all made incredible and sacrifices for me over many years that I might someday have this privilege.

I would like to thank Prof. Badache Nadjib for agreeing to chair my thesis jury, as well as Prof. Benchaïba Mahfoud, Dr. Tandjaoui Djamel, Prof. Slimani Hachem and Dr. El Bouhissi Houda for honouring me with their acceptance as examiners in my thesis jury. I feel extremely lucky and humbled to have had the opportunity to learn from such an impressive and supportive committee.

List of Figures

List of Tables

List of Acronyms

IoT Internet of Things

RMSE Root Mean Square Error

RFID Radio Frequency IDentification

NFC Near Field Communication

WSN Wireless Sensor Network

SOA Service Oriented Architecture

HTTP Hypertext Transfer Protocol

URI Uniform Resource Identifiers

API Application Programming Interfaces

M2M Machine-to-Machine

DTLS Datagram Transport Layer Security

MQTT Message Queuing Telemetry Transport

XMPP Extensible Messaging and Presence Protocol

AMQP Advanced Message Queuing Protocol

AI Artificial Intelligence

ANN Artificial Neural Networks

SSN Semantic Sensor Networks

SBI Sequential Bayesian Inference

ML Maximum Likelihood

KF Kalman Filter

PF Particle Filter

CC Covariance Consistency Model

DS Dempster-Shafer

SMC Sequential Monte Carlo

KLD Kullback-leiber Divergence

BPA Basic Probability Assignment

Contents

Chapter 1

General Introduction

The Internet of Things is an emerging technology in telecommunications and computer networks continuously evolving to fit various domains. This evolution will be accompanied by an evolution of the technological ecosystem in all its complexity. This technology is taking a large part of the computer systems market, notabely in smart city applications, due its flexibility and adaptability in several fields. Each object can autonomous, able to interact and cooperate with other objects in order to achieve common goals. It has a unique address or ID, and should not consume a lot of energy. IoT is considered as a global infrastructure for the information, where it is expected that billions of devices or things that are able of sensing, communicating, computing, and potentially of actuation will be connected to the Internet[1]. This includes sensors, RFID (Radio Frequency IDentification), cell phones, smarter watches, smart glasses, etc. The vision of IoT is to allow these things to be connected anytime, anywhere, with anything or anyone, ideally using any path, any network and any service. This will generate a huge amount of data coming from different sources, which arises the need for effective methods for processing such data[2].

The interconnection of sensing and actuation devices enables sharing platform information in a unified framework, developing a common operating image for innovative applications. this is achieved through large-scale detection, data analysis and information representation using ubiquitous detection and emerging technologies such as edge, fog, and cloud computing. Exchanging data and information while reacting autonomously to the real/physical events of the world and influencing the execution of processes that trigger actions and create services with or without direct human intervention. Regardless of the definition given to the Internet

of Things, this technology is always carried out in three paradigms: a middleware, oriented objects (sensors) and oriented semantics (knowledge). The usefulness of IoT can only be seen in a field of application where the three paradigms intersect. This type of delimitation is necessary because of the interdisciplinary nature of the subject.

A highlight of the IoT is its pervasiveness in all areas of life. The IoT design must be pervasive, scalable, interoperable, consistent, reliable, efficient, and secure. Meanwhile, data that must be supported in the IoT can be multisource, heterogeneous, massive, redundant, inconsistent, or unreliable. Data fusion is an important tool in preparing for the influx of massive amounts of IoT information. Data fusion deals with these problems for achieving situation awareness and enabling applications, machines, and human users to understand each other more fully, provide advanced intelligence, and interact with the dynamics of their environments.

Data fusion is defined as the theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representational format [3]. It is also considered as the combination of information from different heterogeneous sources of measurement [4–6]. The goal in data fusion is to improve the performance of a given system by combining complementary or redundant information. The combination of redundant information makes it possible to reduce the uncertainty of the measurements, whereas the combination of complementary information makes it possible to obtain information that cannot be perceived by a single sensor. Data fusion is commonly used for detection and classification in different application domains [7–9], such as military, robotics, medical, earth sciences, and industrial applications. The fundamental elements of a distributed information fusion system in IoT are the sensors and processors. Sensors are responsible for data generation by observing the operating environment, while processors are responsible for fusing the data. Data fusion in a distributed and heterogeneous environments such as IoT is challenging. Data perceived by various sensors may be imprecise, inaccurate and uncertain due to data loss or data source unreliability, which brings additional challenges for data fusion caused by data imperfection, data conflict, data ambiguity and inconsistency. The system must be energy efficient, secured for both data and technology. Scalability is another challenge in IoT [10] featured with the frequent changes in the shape and size of the networks.

In our work, we will focus on the following question:

How we can ensure the efficiency of data processing in IoT environments, with more focus on data fusion, in terms of credibility, reliability, conflict and time latency?

The rest of chapters are organized as follows:

- Chapter 2 presents some general concepts and background then provides an overview on characteristics, goals, components and challenges of data fusion technique in IoT.

- In Chapter 3 a state-of-the-art of frameworks and platforms of data processing is presented. Moreover, a state-of-the-art on data fusion methods is provided. This allows elaborating a taxonomies of proposed frameworks and improved methods associated to data fusion in IoT environment.

- In Chapter 4 we propose two efficient data processing frameworks to ensure real time data processing and reduce the network energy. The first one is based on edge computing and enables sophisticated services via the Internet in the emerging internet of things. In contrast to the existing approaches, and by processing the data at the edge, the data privacy is preserved and the bandwidth for data relaying is saved. With this architecture, communication overhead can be significantly reduced, and services in the cloud ensure real time data processing. The second solution is an efficient hybrid computing platform for data management and control in smart cities that allows heterogeneous resources, connectivity reliability and mobility, It ensures security and contains services in application framework architecture.

- In Chapter 5 we propose improved distributed particle filter algorithm to deal with target tracking, a new algorithm with new metric. It addresses the measurement uncertainty problem and makes the particle filter robust to environmental change. Such an approach can be used in state estimation to fuse data and applied in smart environments and Internet of things applications. A simulation study that compares the proposed solution with two state-of-the-art solutions shows the superiority of the proposed approach in RMSE (Root Mean Square Error) and ensures scalability.

- In Chapter 6 the problem of data credibility, reliability, and conflict are considered. A new method for data fusion in IoT is proposed in this chapter. We compared the proposed solution with some state-of-the-art data fusion methods, using both benchmark data simulation and real dataset from a smart building testbed. Results show that our method outperforms all the

above mentioned methods in terms of reliability, accuracy and conflict management.

- Chapter 7 summarizes and discusses the research pursued in this thesis and the obtained results. The contributions are reviewed, and reflections on the results are given. The dissertation concludes with an outline of future work.

Chapter 2

Backgroud

2.1 Introduction

As described in Chapter 1, this dissertation focuses on the data processing in the emerging internet of things in terms of integrity control, reliability and time latency. In this chapter, some general concepts used throughout the dissertation are defined. We first give a general overview on IoT, operation, and Protocols. This will be followed by introducing the data processing paradigm in IoT, and the related relevant concepts. We identify two main backgrounds required for the comprehension of the upcoming chapters. The first is related to the IoT architecture and technologies, whereas the second one is related on the data processing. Accordingly, in the second part of the chapter, we present the data fusion methods and application domains as well as the goals and challenges of data processing.
The Internet is changing and evolving. The main form of communication of the current Internet is human-human. IoT can be seen as the future evolution of the Internet that realizes machine-to-machine learning. Thus, the IoT provides connectivity for everyone and any object. IoT incorporates intelligence into Internet-connected objects to communicate, exchange information, make decisions, invoke actions and provide surprising services [1]. The technology of the Internet of Things will act on many areas of human life by facilitating a few and adding comfort to a few others (smart farm, smart home, smart health, smart city, etc.) In this chapter we will introduce the different essential notions of IoT, its architecture, its elements, the fields of application, and the challenges that have emerged with this technology.

2.2 Internet of Things

IoT is the interconnection of detection and actuation devices offering the ability to share platform information in a unified framework, developing a common operating image to enable innovative applications. This is achieved through large-scale detection, data analysis and information representation using ubiquitous detection and cloud computing. According to the RFID group: IoT is the global network of interconnected objects that can be uniquely addressed according to standard communication protocols [2]. According to the cluster of European research projects on the IoT: Objects are active participants in interactions, information and social processes where they are able to interact and communicate with each other and with the environment. Exchange data and information react autonomously to the real/physical events of the world and influence the execution of processes that trigger actions and create services with or without direct human intervention [2]. Regardless of the definitions given to the IoT, this technology is always carried out in three paradigms: a middleware, oriented objects and oriented semantics. The usefulness of IoT can only be seen in a field of application where the three paradigms intersect, indeed this type of delimitation is necessary because of the interdisciplinary nature of the subject [3].

2.3 Smart Objects

Smart objects, also known as intelligent objects, are objects that are equipped with positioning and communication technologies and are integrated into a communication network. These intelligent objects can enter, store and process data and interact with other objects, systems or people. They can be embedded or fixed in other objects and capture data about position and sensors, as well as execute decision and control functions.

2.4 Evolution of IoT

Extracted from the URL

- 1999: The term "Internet of Things" is coined by Kevin Ashton, Executive Director of the Auto-ID Center Massachute Institute of Technology.

- 1999: Neil Gershenfeld for the first time mentions the principles of IoT in his book "When Things Start to Think" [11].
- 2000: LG announces its first smart refrigerator.
- 2002: The Ambient Orb created by David Rose and others in a spin-off of the MIT Media Lab published in NY Times Magazine named as one of the ideas of the year.
- 2003-2004: RFID is deployed massively by the US Department of Defense in their Savi program, and Wal-Mart in the commercial world.
- 2005: The United Nations International Telecommunication Union publishes his first report on the Internet of Things.
- 2008: After being recognized by the European Union, the first European Conference on the Internet of Things took place.
- 2008: A group of companies launched the IPSO Alliance to promote the use of IP in the networks of "Smart Objects" and enable the Internet of Things.
- 2008: The FCC voted 5-0 for the opening of the "white space" spectrum.
- 2008-2009: IoT was born according to Cisco's Business Solutions Group.
- 2008: The US National Intelligence Council has identified IoT as one of six "disruptive civilian technologies" that have potential repercussions on the interests of the United States until 2025.
- 2010: Chinese Premier Wen Jiabao calls IoT a key industry for China and intends to make significant investments in the Internet of Things.
- 2011: launch of IPv6, the new public protocol allowing 2^{128} addresses.

2.5 IoT Architecture and Operation

2.5.1 Layered Architecture

Essentially the structure of IoT [4] is divided into five layers named :

2.5.1.1 Perception Layer

This is the interface layer with the environment. It includes the physical layer and the data access layer of the open systems interconnection model. It serves to identify the objects and collect the information through the sensors. The information can be a measure of geographical position, temperature, humidity, or vibration, depending on the type of sensor used. The collected information is passed to the network layer for transmission to the processing system.

2.5.1.2 Network Layer

Also called transmission layer, it ensures the transmission of information collected by the sensors to the processing system. The transmission may be wired or non-wired depending on the technology used (3G, WiFi, Blueutooth, ZigBee, etc.) Thus the network layer is the intermediary between the perception layer and the middleware layer.

2.5.1.3 Middleware Layer

In IoT each object implements different services, and it only communicates with objects implementing the same type of services. The middleware receives information from the network layer and stores it in a database, applies processing and calculations to make decisions automatic. The main objective of the middleware layer is to offer the developer a level of abstraction, allowing the implementation of new services and the integration of new technologies without considering those used in the lower layers.

2.5.1.4 Application Layer

Allows the user to fully manage applications based on the information processed in the middleware layer. Applications can be different domains (health, transport, home automation, agriculture, etc.)

2.5.1.5 Business Layer

Responsible for the overall management of the system as well as the applications and services. It is used to develop business models, graphs, flowcharts, based on

the information received from the application layer. Using the results analysis, this layer helps establish effective business strategies. The layered architecture is described in Fig. 2.1

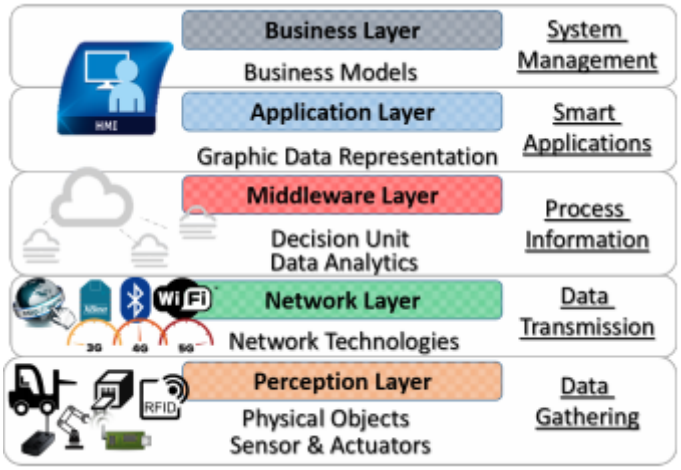


FIGURE 2.1: IoT layered architecture

2.5.2 Addressing

The IoT will include an incredibly high number of nodes, each of which will produce content that can be retrieved by an authorized user regardless of his position. This requires effective addressing policies. currently, the IPv4 protocol identifies each node via a 4-byte address. It is well known that the number of IPv4 addresses available is rapidly decreasing and will soon reach zero. Therefore, it is clear that other addressing policies should be used other than those used by IPv4. The most critical features of creating a unique address are: uniqueness, reliability, persistence, and scalability [7]. IPv4 can support to the extent that a group of cohabiting sensors can be identified geographically, but not individually, IPv6 comes with addresses that are expressed by means of 128 bits, which is sufficient for identification of billions of objects on a global scale.

2.5.3 Storage and Processing

One of the most important results of this emerging field is the creation of amounts of data. Storage and expiration of data become critical issues. The data must be stored and used intelligently for monitoring and intelligent actuation. It is important to develop artificial intelligence algorithms that could be centralized or

distributed as needed. New fusion and learning methods must be developed to make sense of the data collected and to achieve automated decision making.

2.5.4 Visualization

Visualization is essential for an IoT application, as this allows user interaction with the environment. With the recent advances in touch screen technologies, the use of tablets and smart phones has become very intuitive. To fully benefit from the IoT revolution, an attractive and easy-to-understand visualization must be created. As we move from 2D to 3D screens, additional information can be provided to the user in a meaningful way for the consumer. It will also enable policy makers to convert data into knowledge that is essential for rapid decision-making. The extraction of meaningful information from raw data is not trivial. This includes both event detection and visualization of raw and modeled data, associated with information represented according to end user needs.

2.6 IoT Technology and Protocols

2.6.1 Perception Layer

2.6.1.1 Sensors

Sensors are small, energy-efficient electronic devices used to monitor a specific change and transmit it to actuators and the database as useful information. In the literature we find many types of sensors (temperature, movement, light, etc.), as shown in Fig. 2.2.

2.6.1.2 Actuators

An actuator converts energy into motion, which means that actuators cause movements in mechanical systems. Hydraulic fluid, electric current, or other power source is required. Actuators can create linear motion, rotary motion, or oscillating motion. Cover with short distances, typically up to 30 feet, and generally communicate at less than 1 Mbps. Actuators are typically used in manufacturing or industrial applications. There are three types of actuators: (i) Electric: AC

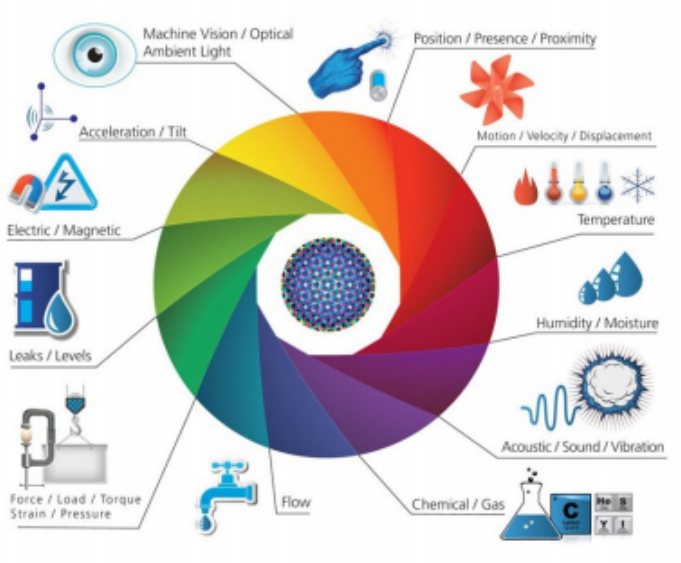


FIGURE 2.2: Sensors

motors, stepper motors, solenoids. (ii) Hydraulic: use a hybrid fluid to actuate the movement. (iii) Pneumatic: use compressed air to activate the movement. All of these three types of actuators are widely used today [5].

2.6.1.3 RFID

RFID is a system that provides wireless transmission of the identity of an object or person using radio waves in the form of a serial number. RFID technology plays an important role in the IoT to solve the problems of identifying objects around us in a cost-effective manner. The technology is classified into three categories based on the method of supplying power in the labels: Active RFID, passive RFID and semi-passive RFID [5]. Usually attached to an antenna that looks like a regular sticker. The microchip itself can be as small as a grain of sand, around 0.4mm2. An RFID tag transmits data over the air in response to an interrogation by an RFID reader [6]. It is more reliable, efficient, secure, inexpensive and very precise. RFID has a wide range of wireless applications such as distribution, tracking, patient monitoring, military applications.

2.6.1.4 NFC

NFC (Near Field Communication) defined as a promising short range wireless communication technology that facilitates the use of mobile phone by offering

various services ranging from transaction payment applications, digital content exchange, etc. NFC technology was jointly developed by Philips and Sony at the end of 2002 for wireless communications [8]. It is a short range communication protocol, which provides easy and secure communication between different devices. NFC is distinct from radio frequency communication which is used in personal field and long range wireless networks [3]. NFC relies on inductive coupling between sending and receiving devices. Communication occurs between two compatible devices within a few centimeters with an operating frequency of 13.56 MHz. The data exchange rate is around 424 kbps.

2.6.1.5 WSN

WSN (Wireless Sensor Network) [9] consists of a number of sensor nodes working together to monitor an area to obtain data on the environment. There are two types of WSN: unstructured and structured.

- An unstructured wireless sensor network is one that contains a dense collection of sensor nodes. Once deployed, the network is left unattended to perform monitoring and reporting functions. In a WSN unstructured, network maintenance, such as connectivity management and fault detection, is difficult because there are so many nodes.
- In a structured wireless sensor networks all or part of the nodes sensors are deployed in a pre-planned manner. The advantage of a structured network is that fewer nodes can be deployed with maintenance and a cost of network management lower. Fewer nodes can be deployed as compared to nodes are placed in specific locations to provide coverage while ad hoc deployment may have regions not covered.

2.6.2 Network Layer

2.6.2.1 IEEE 802.15.4

The IEEE 802.15.4 protocol was created to specify a sub layer for MAC layer, and a physical layer for low speed wireless broadband networks [7]. Due to its specifications such as low power consumption, low data rate, low cost and high message throughput, it is also used by IoT, M2M and WSN. It provides reliable

communication on different platforms, and can handle a large number of nodes. It also offers a high level security, encryption and authentication services. However, it does not provide any guarantees of quality. This protocol is the basis of the ZigBee protocol because they both focus on low data rate services on devices with power constraints, and they create a complete network protocol stack for WSNs. IEEE 802.15.4 supports three bands of frequency chains and uses a direct sequence spread spectrum method. Based on the frequency channels used, the physical layer transmits and receives data at three data rates: 250 kbps at 2.4 GHz, 40 kbps at 915MHz and 20 kbps at 868MHz. Higher frequencies and wider bands provide high throughput and low latency while frequencies lower ones provide better sensitivity and cover greater distances. To reduce potential collisions, the IEEE 802.15.4 MAC uses the CSMA/CA protocol.

2.6.2.2 Bluetooth

Bluetooth is a standard for wireless communications based on a radio system designed for short range, inexpensive communication devices suitable for replace cables for printers, fax machines, keyboards, etc. The devices could also be used for communications between laptops, act as bridges between other networks, or serve as nodes of ad hoc networks. This range of applications is known as wireless personal area network.

2.6.2.3 WiFi

WiFi (Wireless Fidelity) technology uses radio waves to communicate data between objects within a range of 100m [7]. It allows peripherals devices to communicate and exchange information without using a router in some ad hoc configurations, and is governed by the IEEE 802.11 group standards (ISO/IEC 8802-11).

2.6.2.4 ZigBee

The ZigBee Alliance is an association of companies working together to develop standards and products for reliable, cost-effective, low-cost wireless networks power. ZigBee is based on the IEEE 802.15.4 standard which defines the physical and MAC layers for networks with low energy consumption [10]. ZigBee defines the network layer specifications for star topologies, tree and peer-to-peer and provides

a framework for programming applications in the application layer. The main characteristics of a standard ZigBee device are:

- Low data rate (maximum 127 bytes/s).
- Low power (usually uses 2 AA batteries for up to 2 years)
- Low price.
- Uses three frequencies: 868, 915 MHz and 2.4 GHz.
- low bandwidth (250 kbps in the 2.4 GHz band).
- Supports three network topologies (star, tree, mesh).
- Support for a large number of modes in the network.
- Uses ad hoc networks.
- Quick establishment of connections.
- Support for a large number of network nodes.
- Support for built-in AES-128 encryption and authentication.

2.6.2.5 6LoWPAN

6LoWPAN is an acronym of IPv6 over Low Power Wireless Personal Area Networks. The 6LoWPAN concept originated from the idea that "the Internet Protocol could and should be applied even to the smallest devices", and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things. The 6LoWPAN group has defined encapsulation and header compression mechanisms that allow IPv6 packets to be sent and received over IEEE 802.15.4 based networks. The use of 6LoWPAN on top of IEEE 802.15.4 provides for secure, transparent connectivity with the cloud and significantly reduces the burden on developers and system designers by providing standard IP compatible protocols and readily available libraries [12].

2.6.3 Middleware Layer

2.6.3.1 Architecture

Middleware is a software layer or a set of sub-layers interspersed between the technological and application levels [3]. Middleware connects different, often complex and already existing programs that were not originally designed to be connected. The essence of the Internet of Things is making it possible for just about anything to be connected and to communicate data over a network. Middleware is part of the architecture enabling connectivity for huge numbers of diverse Things by providing a connectivity layer for sensors and also for the application layers that provide services that ensure effective communications among software

2.6.3.2 SOA

SOA (Service Oriented Architecture) can be both an architecture and a programming model, a way of thinking about building software. SOA architecture allows to design software systems that provide services to other applications through published and discoverable interfaces and where the services can be invoked over a network. The implementation of SOA architecture using web service technologies creates a new way to implement applications in a more powerful and flexible programming model. The middleware architectures proposed in recent years for the IoT often follow the SOA approach [3], The adoption of SOA principles makes it possible to break down complex and monolithic systems into applications composed of a ecosystem of simple and well-defined components. An SOA approach also enables reuse of software and hardware, as it does not impose specific technology for the implementation of the service.

2.6.4 Application Layer

2.6.4.1 REST Architecture

REST (Representational State Transfer) is an abstraction of the architectural elements of a distributed hypermedia system. REST is independent of component implementation details and protocol syntax [13]. It focuses on the roles of components, the constraints on their interaction with other components, and their

interpretation of meaningful data elements. It encompasses the fundamental constraints on the components, connectors, and data that define the basis of the web's architecture and thus the essence of their behavior as a network application.

Architectural Elements: REST uses various types of connectors to encapsulate the resource access and transfer activities of resource representations. REST components are typed by their roles in an action overall application. A user agent uses a client connector to initiate a request and becomes the final recipient of the response. The original server is the definitive source of representations of its resources, it must be the final receiver of any request that intends to modify the value of its resources. It provides a generic interface to its services via a hierarchy of resources. The details of the implementation of the resource are hidden behind the interface. A proxy component is an intermediary chosen by a client to provide an interface encapsulation to other services, data translation, improvement performance or safety protection.

The Constraints Applied: The architecture client-server improves the portability of the user interface, the ability to scale out and allows components to scale independently. The fundamental point that distinguishes the REST architecture model from other models based on network concepts is the emphasis on a uniform interface between the components. By applying the software principle of generalization to the component interface, the overall architecture of the system is simplified and the visibility of interactions is improved. The layered system model allows an architecture to be composed of hierarchical layers by constraining the behavior of the components. Each component cannot see beyond the immediate layer with which it interacts. REST allows the extension of a client's functionality by downloading and executing code in the form of applets or scripts. The ability to download features after deployment improves system scalability. However, it reduces visibility. Therefore, it constitutes an optional constraint in REST.

The Standards Used:

HTTP(Hypertext Transfer Protocol) is an application layer protocol, intended for client/server communications, hereafter the most used methods:

- POST: used for the creation of new resources.

- GET: used for resource recovery.
- PUT: used to perform a replacement update.
- PATCH: unlike PUT it is used for updating and in partial modifications.
- DELETE: useful for resource deletions.

URI(Uniform Resource Identifiers) a character string that uniquely identifies the resources of the WEB. A URI can be of type locator or name or both such as:

- A Uniform Resource Locator is a URI which, in addition to the fact that it identifies a resource on a network, provides the means to act on this resource or to obtain a representation of it by describing its mode of operation. primary access or network location.
- A Uniform Resource Name is a URI which identifies a resource by name in a namespace without prejudging its location or the way it is referenced.

Hypermedia Links in HTML and XML documents to represent both information content and the transition between application states.

2.6.4.2 CoAP

The use of web services or web APIs (Application Programming Interfaces) on the Internet has become ubiquitous in most applications and depends on the REST from the web [7]. **Co**nstrained **A**pplication **P**rotocol is a web transfer protocol specialized intended for use with constrained nodes and restricted networks. Nodes often have 8 micro controllers bits with small amounts of ROM and RAM, while restricted networks like IPv6 over 6LoWPAN often have High packet error rates and a typical throughput of 10s of kbit/s. The protocol is designed for M2M (Machine-to-Machine) applications such as smart energy and building automation. Many data fusion method in the literature use this protocol [14] in IoT environment. Fig. 2.3 shows the CoAP functionality.

CoAP has the following main characteristics:

- Web protocol meeting M2M requirements in constrained environments.

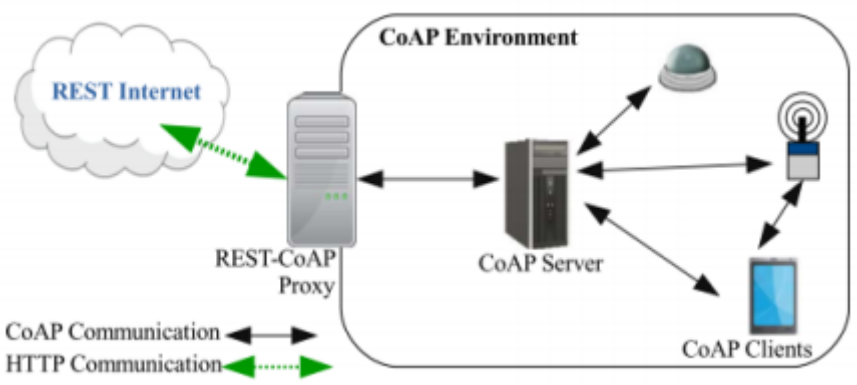


FIGURE 2.3: CoAP functionality

- UDP link with optional reliability supporting unicast requests and multicast.
- Asynchronous message exchanges.
- Low header overhead and analysis complexity.
- URI and content type support.
- Simple proxy and caching capabilities.
- A stateless HTTP mapping, allowing to build proxies giving access uniformly to CoAP resources via HTTP or to achieve interfaces simple HTTP alternately over CoAP.
- Security link to DTLS (Datagram Transport Layer Security).

2.6.4.3 MQTT

MQTT (Message Queuing Telemetry Transport) is a messaging protocol that was introduced by Andy Stanford-Clark IBM and Arlen Nipper from Arcom (now Eurotech) in 1999 and was standardized in 2013 at OASIS. MQTT aims to connect on-board devices and networks with applications and middleware. The connection operation uses a routing mechanism (one-to-one, one-to-many, many-to-many) and allows MQTT protocol as protocol optimal connection for IoT and M2M. MQTT uses the publish/subscribe model to provide flexibility of transition and simplicity of implementation [15], MQTT is suitable for resource-limited devices using unreliable links or low bandwidth. MQTT is built on the TCP protocol, it delivers messages to through three levels of QoS (sent and forgotten, delivered at least once, delivered exactly once). MQTT consists of three components: subscriber,

publisher, and broker. An interested device signs up as a subscriber to specific topics so that they will be notified by the broker when publishers post topics of interest. The editor acts as a generator of interesting data. After that, the publisher transmits the information to interested entities (subscribers) through the broker. In addition, the broker achieves security by verifying the authorization of publishers and subscribers.

2.6.4.4 XMPP

XMPP (Extensible Messaging and Presence Protocol) was designed to chat and message exchange. It has been standardized by the IETF (Internet Engineering Task Force) [16]. XMPP operates over TCP and provides publish / subscribe (asynchronous) and request/reply (synchronous) messaging systems. It is designed for near real-time communications and therefore supports low message congestion and low latency message exchange. XMPP has a TLS/SSL security built into the core of the specification. However, it does not provide QoS options that make it impractical for M2M communications. Only the Mechanisms inherited from TCP ensure reliability. XMPP uses XML (eXtensible Markup Language) messages which generate additional overhead due to unnecessary tags and require XML parsing which requires additional computing capacity and increases energy consumption.

2.6.4.5 AMQP

The AMQP (Advanced Message Queuing Protocol) is a protocol of the financial sector. It can use different transport protocols but it assumes a reliable transport protocol like TCP. AMQP provides asynchronous publish/subscribe communication with messaging. Its main advantage is its store and forward function which guarantees reliability even after network interruptions [16]. It ensures reliability with following message delivery guarantees:

- At most once: means that a message is sent once whether it is delivered or not.
- At least once: means that a message will be definitively delivered once, maybe more.
- Exactly once: means that a message will only be delivered once.

Security is managed with the use of TLS/SSL over TCP protocols.

2.6.4.6 WebSocket

The Websocket protocol was developed as part of the HTML5 initiative to facilitate communication channels over TCP. Websocket is neither a request/response nor a publish/subscribe protocol. In Websocket, a client initiates a negotiation with a server to establish a Websocket session [16]. The handle from main itself is similar to HTTP so that web servers can handle Websocket sessions as well as HTTP connections through the same port. However, what comes after the handshake does not comply with HTTP rules. Indeed, during a session, HTTP headers are removed, clients and servers can exchange messages in an asynchronous full-duplex connection. The session can be stopped when no longer needed from either the server or the client. Websocket runs over trusted TCP. If necessary, sessions can be secured using Websocket over TLS/SSL. Websocket is designed for real-time communication, it is secure and minimizes overhead costs.

2.7 Data Processing in IoT and some Relevant Concepts

IoT is a network of devices and objects that are connected to the Internet. Being connected to the Internet means that they can either collect data and send it through the Internet, receive information from the Internet, or do both the things. All solutions in IoT typically involves four components : sensors, connectivity, data processing, and a user interface.

Sensors are objects that collect data and send it over the Internet. The data could be sent for storing, processing, or further dissemination of information.

Connectivity is the piece of the IoT puzzle which enables the things to communicate and exchange data. The connection can be achieved via wired or wireless network. However, wired network is unsuitable for most IoT applications because its range is only as far as the wire can reach.

The amount of data collected by IoT devices is humongous. The amount of storage space as well as processing capacity required to utilize this data is also very huge.

Solutions for both storage and processing are proving to be benefit because they are affordable, scalable, fast response times, and quick time to market.

The user interface consists of the features by which a user interacts with a computer system. This includes screens, pages, icons, forms, etc. The most obvious examples of user interfaces are software and applications on computers and smartphones.

All the four components of an IoT solution are important but data processing proves to be the most challenging as well as crucial.

2.7.1 Definition

The volume and pace at which data is produced nowadays is unbelievable. About 90% of all data in the world today has been produced just in the past two years. In order to make sense of the massive amount of data our IoT sensors collect, we need to process it. Wikipedia explains data processing as the collection and manipulation of items of data to produce meaningful information. In other words, the purpose of data processing is to convert raw data to something useful. Something the end user can react to. We should also take notice of the difference between data and information. Data refers to raw, unorganized facts, and it usually is fairly useless until it is processed. Once the data is processed, it is called information.

2.7.2 Data Processing Cycle

Data processing in IoT follows the typical Input>Process>Output cycle of any computer activity, as shown in Fig. 2.4.

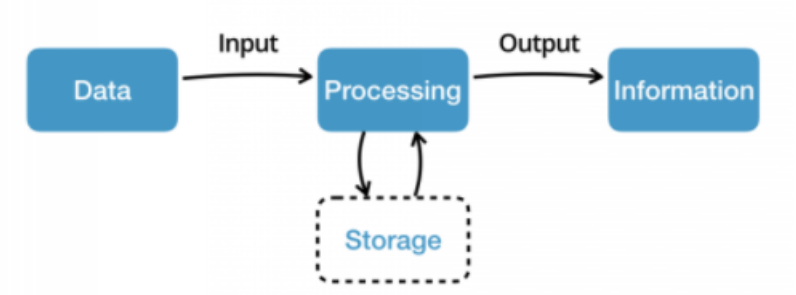


FIGURE 2.4: Data processing cycle

2.7.2.1 Input

Input is the first stage of the data processing cycle. It is a stage in which the collected data is converted into a machine-readable form so that a computer can process it. This is a very important stage since the data processing output is completely dependent on the input data. The data collected may be in the form of images, QR codes, text, numbers, or even videos. All these data must be converted into machine readable form before they can be sent for processing.

2.7.2.2 Processing

In the processing stage, a computer transforms the raw data into information. The transformation is carried out by using different data manipulation techniques like classification, fusion, calculation, etc. to get meaningful information from the data received.

2.7.2.3 Output

Although the information is produced in the processing phase itself, it is rendered into human-readable format in the output stage. This output maybe in the form of graphs, tables, audio, video, etc. Output may also be stored as data for further processing at a later date. This is essential because comparison of current information with historical data can produce useful insights into the overall functioning of a system. This comparison can also be used to predict future behavior.

In this thesis, we will concentrate on the concept of data fusion which is indispensable part in data processing.

2.8 Data Fusion in IoT

Data fusion techniques combine multiple data sources to obtain improved information (cheaper, higher quality, or more relevant/useful information)[[17],[18]]. Data fusion is a data processing technique that associates, combines, aggregates, and integrates data from different sources. It helps to build knowledge about certain events and environments which is not possible using individual sensors separately

An enormous amount of data is produced in a quick span of time in the IoT environment [19]. How to make this large volume of data precise and highly accurate is an open problem which needs to be solved because the quality of information plays an important role in decision making. Reliable and accurate information is critical. This can be achieved by data fusion. Data fusion is an effective way for the optimum utilization of large volumes of data from multiple sources [20]. Multi-sensor data fusion seeks to combine information from multiple sensors and sources to achieve inferences that are not feasible from a single sensor or source [21]. The fusion of information from sensors with different physical characteristics enhances the understanding of our surroundings and provides the basis for planning, decision-making, and the control of autonomous and intelligent machines.

2.8.1 Classification of Methods

There are several classifications of data fusion methods. Here we present the most used and the most frequent.

1. Focus on relationship among the data sources [22]. (i) Complementary fusion involves fusing different portions of a picture into a more complete picture (for instance, temperature readings from different locations within a greenhouse in order to better understand the temperature status of the greenhouse as a whole). In contrast, (ii) redundant fusion combines multiple instances of the same information to increase reliability, accuracy, and/or confidence (e.g., multiple temperature readings from the same location in a greenhouse to get a more reliable or accurate understanding of the temperature at that location). Finally, (iii) cooperative fusion combines multiple independent sources of information into new, more complex information (e.g., temperature combined with light yields a better understanding of overall growing conditions within the greenhouse).

2. Depending on the type of architecture [23]: (i). Centralized in which the merge node resides in the central processor which receives information from all input sources in the form of metrics. (ii). Decentralized where each node fuses its local information with the information that is received from its peers. Decentralized data fusion algorithms typically communicate information using the Fisher and Shannon measurements. (iii). Distributed

architecture where measurements from each source node are processed independently before the information is sent to the fusion node. However, the core of data fusion does not lies in its architecture; it indisputably lies in the data fusion methods on which ultimate fusion processing takes place.

3. Based on mathematical methods [4]: (i). Probability-based methods including Bayesian analysis, statistics, and recursive operators. (ii). Artificial Intelligence (AI) based techniques including classical machine learning, fuzzy Logic, Artificial neural networks (ANN) and genetic evaluation. (iii). Theory of Evidence based Data Fusion methods.

4. A general classification based on the similarity of functioning of the methods, proposed by [23]: (i). Data association. (ii). State estimation. (iii). Decisional methods.

2.8.2 Data Fusion Applications in IoT

As we sought more specific and recent examples of academic research and experiments with data fusion methods, we turned to the application areas in which the work is generally published, such as smart cities and transportation, public health, Military, industrial manufacturing and agriculture, and localization. Additional related surveys, as well as individual examples, are covered as depicted in Fig. 2.5.

2.8.2.1 Smart Cities and Transportation

Smart cities are incorporated into the landscape across a wide spectrum of technologies and are designed to effectively integrate urban areas for communication, power management, resource management, transportation, emergency services, law enforcement, and many more applications. IoT networks in smart cities further extend this to citizens through smart offices and buildings, and even mobile devices for pedestrian management [24]. The term smart city encompasses a wide variety of technologies and applications exploited to support added-value services for the administration of the city and for the citizens, as well as data fusion technique applied to extract higher level information or increase the data completeness. The end product of data fusion include visualization of information for respective

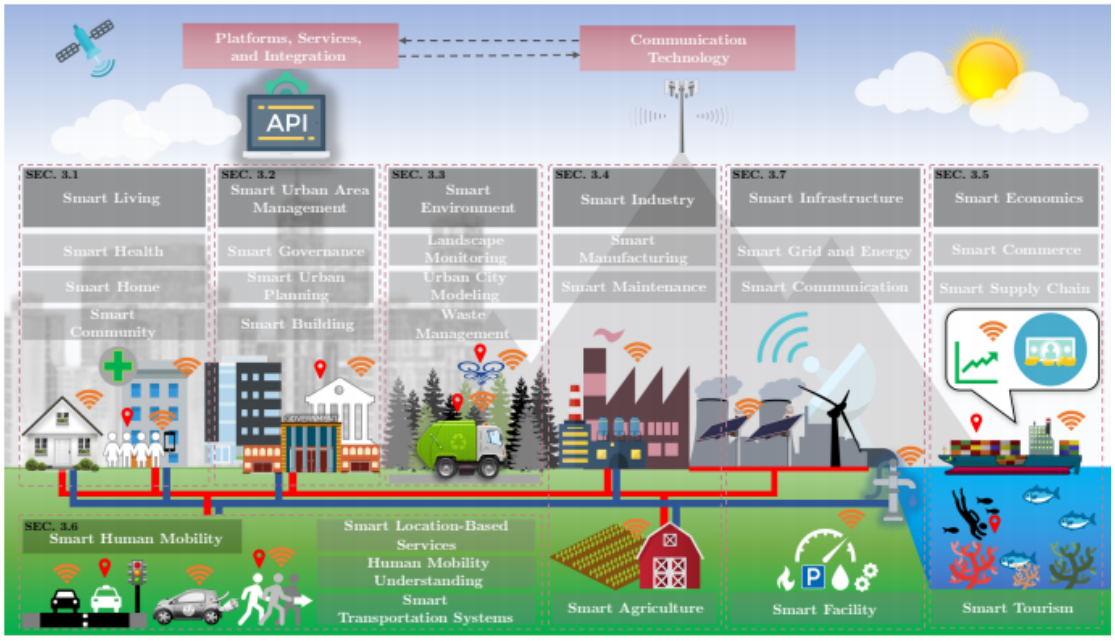


FIGURE 2.5: Application domain

administrations [25]. In intelligent transportation systems [26], transportation infrastructure is complimented with information and communication technologies with the objectives of attaining improved passenger safety, reduced transportation time and fuel consumption and vehicle wear and tear. Data fusion is used to reach a better inference.

2.8.2.2 Health

Data fusion in IoT can be used in health care domain to monitor the state of patients at the hospital or at home. Smart Healthcare can be seen as a complex ecosystem of smart spaces (e.g. hospital rooms, ambulances, pharmacies, etc.), supported by a powerful infrastructure stack including edge devices and sensors, wired and wireless networks, cloud platforms, and driven by innovative business models and legislation enabling the Healthcare Industry. The existing approaches to enable IoT data fusion including the Smart Healthcare domain and processing have primarily adopted either a cloud-centric model [27], where raw data collected by edge devices are pushed to a cloud that is seen as the primary processing location, or hierarchical data fusion approach [28] for timely decision taking in digital healthcare.

2.8.2.3 Military

As the most of existing technology in the world, the military domain is the initiator engine of WSN. Military and defense services use data fusion technique in ocean surveillance, air-to-air or ground-to-air defense, battlefield intelligence, data acquisition, warning, defense systems, etc., using EM radiation from large distances [29]. The technique has been deployed on several major military weapons systems, such as the U.S. Navys Cooperative Engagement Capability, a system that enables Navy ships and aircraft to combine radar data for improved defenses against attack aircraft and cruise missilesparticularly in coastal waters where land clutter can make it difficult to formulate a reliable radar picture.

2.8.2.4 Industrial Manufacturing

data fusion of smart factory sensors allow for a myriad of improvements in production. They can control energy and water usage to reduce waste and optimize for environmentally sustainable operations. Quality control in the smart factory can be enhanced with real time analytic on the supply chain [30]. The exchange and combination of information from other systems and devices directly back into the production line, enables predictive maintenance and forecasts needed.

2.8.2.5 Agriculture

In agricultural domain , sensors use both predictive approaches and control approaches that include information from static indicators. As with industrial manufacturing. Smart farming has been expanded by introducing cloud computing and IoT solutions [31]. Using cloud computing and key IoT techniques, visualization and SOA technologies can take advantage of massive data involved in agricultural production.

2.8.2.6 Localisation

The localisation problem including indoor and outdoor spans multiple fields such as physics, sensor fusion, and real-time computation. The indoor localisation problem is more complex than just finding whereabouts of users. Finding positions of users relative to the devices of a smart space is even more important. The recent studies propose a system to address the problem of locating devices and users relative to

those devices [31], and employ data fusion techniques to solve this problem using motion data from users.

2.9 Data Processing Challenges in IoT

2.9.1 Big Data

Big data processing is a research area that ensures the aggregation of data generated either independently or collectively. It facilitates an improvement in decision making through value extraction. The result of this is data are better generated, stored, manipulated and analyzed [32]. But it also faces a number of challenges caused by limited storage and energy of sensor devices. It is also difficult to distinguish spurious data, and wide distribution of this data incurs high communication and serious delay in data processing models.

2.9.2 Heterogeneous

Heterogeneity in IoT environment is a challenging issue to handle during system integration due to different sources of data. The involvement of data from various sensors enhances the breadth of collected data. But the diversity of data with different types, forms, representations, scales and densities makes it hard to fuse the data directly. Methods are required to transform heterogeneous data to homogeneous space [33]. Further heterogeneous datasets add uncertainty. Complex multivariate relationships among the datasets. However, processing data from heterogeneous observations promises to find complex multivariate relationships among the data sets.

2.9.3 Data Quality

Quality of the data sources directly determine the quality of output results since processing module follows the GIGO (garbage in and garbage out) theorem in fusing data sources [25]. Data perceived by various sensors may be imprecise, inaccurate and uncertain due to data loss or data source unreliability, which brings additional challenges for data fusion caused by data imperfection, data conflict, data ambiguity and inconsistency.

2.9.3.1 Data Imperfection

Sensor data is imprecise at times; it can be inaccurate and uncertain. This behavior is not infamous in wireless sensor networks. The imperfection must be dealt with effectively with the use of data fusion algorithms.

2.9.3.2 Ambiguities and Inconsistencies

Impreciseness is not the only factor responsible for data inconsistencies; the environment in which a sensor is operating is largely responsible as well. Outliers detection, replacement and data imputation are vital in IoT environment.

2.9.3.3 Conflicting Nature

The conflicting nature of data can give rise to counter-intuitive results. The problem of conflicting data is visible more in evidential belief reasoning and Dempsters rule of combination. The data fusion algorithm must take critical care while treating conflicting data.

Another challenges should be considered in the process of data fusion such as triviality and correlation data, and most relevant features need to be selected before that process.

2.9.4 Energy Saving

Energy efficiency plays a critical role in data processing and also in IoT environment [34], since we have hundreds of sensors operating together. The system must be energy efficient. Otherwise, a lot of cost is incurred on energy consumption by the sensors, the different data processing methods will be discussed and evaluated in terms of energy consumption.

2.9.5 Security

Processing multisensory data increases the risk of privacy invasion. How to preserve user privacy during data processing and at the same time ensure accuracy is an important research issue. There are two general security concerns data fusion

in IoT environment, which are security of technology/infrastructure (data center, services, and system architecture) and data security (data generation, storage, and communication). The security of the technology and infrastructure highly relies on the design architecture of the system being deployed [35]. The main objective is to deploy a hack-proof/exploit-less system architecture. Alternately, propose different strategies to enhance the security of such architecture by focusing on the common security standards/practices/protocols. Meanwhile, data security also contributes to the significant part of applications ecosystem. Observed data may always carry some personal information of an observed target. The common method to combat such issue is leveraging encryption techniques [36], where it encodes the data so that only the authorized parties have access to it. there is also a need for distributed blockchain technologies for the failure and risk-free computation of IoT sensor data [37].

2.9.6 Real Time Data

One key challenge of data processing in IoT is performing low-latency analysis with real-time data. For instance, real time applications (medical emergencies,target tracking) require consideration of additional constraints and requirements, to perform the transformation of raw sensor data into more valuable and insightful information in real-time [38].

2.10 Conclusion

In this chapter, we have presented the general concepts that required to understand the rest of the chapters. Moreover, the main challenges of a data processing in IoT environment have been presented. We have explained the IoT architecture and several technology and protocols used during the process of data. In this chapter, we have presented the classification of a data fusion methods as well as the application areas in IoT environment, The various application areas covered in this section offer a broad array of data fusion methodologies, including techniques for fusing both homogeneous and heterogeneous data. The data fusion methods would be presented in the next chapter.

Chapter 3

Data Fusion Frameworks and Methods in IoT

3.1 Introduction

As we have seen in the previous chapters, data fusion provides many advantages for data processing and analytic by enhancing dataset quality and reducing the volume of data transmission. However, the characteristics of IoT data come up with new challenges for data fusion in IoT. First, the large amount of data collected in IoT makes data fusion difficult, which increases the complexity and even introduces inconsistently and conflicted data. Second, the multi-modality and heterogeneity of data collected from various sensors further complicate data fusion. Third, collected data will be transmitted over the Internet, which may undermine privacy and be vulnerable to false data injection. The present chapter gives a taxonomies on the data fusion frameworks and approaches. Data fusion platform aims to update and adapt the dynamic changes in the IoT systems surroundings, whereas data fusion approaches aims to achieve better reliability, accuracy, position estimation, velocity measurement, attribute evaluations and identity exploration. Data fusion solutions consider, besides data imprecise, energy consumption and scalability as critical metrics.

In recent years, the adoption of emerging technologies has revolutionized cloud computing, fog computing and edge computing towards IoT sensor data fusion. These enabling technologies provide a pervasive, reliable and convenient platform to handle IoT sensor datas dynamic, heterogeneous nature. As such, the data

fusion layer aims at developing smart functionality to address a wide variety of IoT-based applications. The objectives of these frameworks are to reduce the computation and storage cost, improve network transmission reliability, reduce the network delay, enhance IoT network security and privacy, ensure scalability, and allow failure and risk-free IoT solutions. In Section 3.2, we focus on the frameworks of the data processing, and classify them according on the architecture into two categories: middleware and application solutions. In the other side, the methods that can add credibility to the data fusion process are mathematical methods. These methods could be characterized as probabilistic, statistical, knowledge-based, inference and reasoning methods. The probabilistic methods include Bayesian networks, maximum likelihood estimation methods, inference theory, Kalman filtering, etc. Statistical methods include covariance, cross variance, and other statistical analyses [39]. Knowledge-based methods include artificial neural networks, fuzzy logic, genetic algorithms, etc. Depending on the problem specification, the appropriate data fusion methods are to be chosen. In Section 3.3, Data fusion solutions are classified according to fusion level into three categories: (i) measurement methods; (ii) characteristic methods; and (iii) decision methods. Further, each category is sub-classified based on its techniques and objectives. Some data fusion method aim to enhance data quality such as reliability and accuracy, whereas the others aim to reduce data latency.

3.2 Data Processing Frameworks in IoT

The IoT sensor network based applications involve dynamic factors, distributed services, and real-time responsive mechanisms. Hence, there is a middleware layer requirement between IoT-based applications and the underlying IoT sensor data. Further, the scalability addresses huge volumes of data that are obtained from the IoT sensor network. To tackle dynamic, real time data processing and scalability issues, solutions through the integration of IoT sensor networks with other emerging technologies, such as cloud computing, fog computing, and edge computing are required.

3.2.1 Classification and Taxonomy

Many frameworks have been proposed in the literature. Depending on the type architecture, we classify data fusion frameworks into two categories: middleware

solutions that provide the reusable functionalities required to meet complex customer requirements, and application systems where data fusion framework solution are narrowly focused on one specific domain. Furthermore, we refine this classification into four categories, depending on the model of computation platform of processed data: edge computing, fog computing, cloud computing, and hybrid computing, as shown in Fig. 3.1

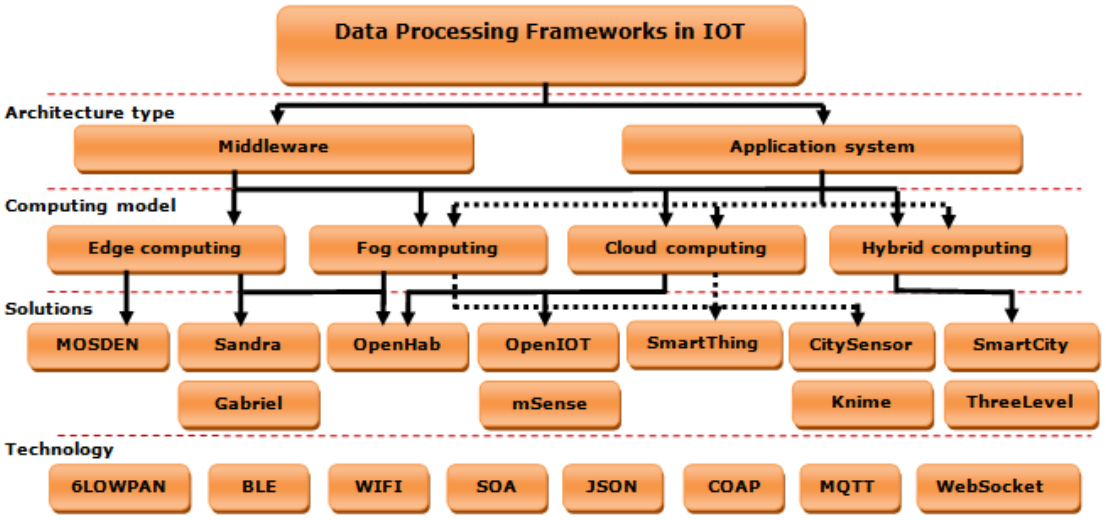


FIGURE 3.1: Taxonomy of frameworks data fusion in IoT

3.2.1.1 Computing Model

Edge Computing In edge computation platform, data sources are processed and fused at the edge which means very close to the physical location, where data is actually collected. Edge computation devices include micro-controller, computing devices (Raspberry pi), computers, etc. Such architecture can be found in works such as [40], [41]. Edge computing can enable real-time data processing with negligible latency due to the close distance between the sensor node and the edge server. This is a desirable property for time sensitive applications like autonomous vehicles. Besides, by processing the data at the edge, the data privacy is preserved and the bandwidth for data relaying is saved. With this architecture, communication overheads can be significantly reduced. However, data processing in this level has limitations, because in-network devices such as sensor nodes and mobile phones perceive only limited knowledge about the environment (local context). Therefore, data processing cannot make high level decision where overall knowledge is required.

Fog Computing The new paradigm of fog computing will serve as the central system architecture principle as it fulfills the requirements of fast data processing. In fog computation platform, data sources are processed and fused at the middle layer, which means between the edge and the cloud. In this architecture, data is periodically or continuously sampled at the edge without processing and is then forwarded to a gateway that acts as a fog device. At the gateway, computing resources are provided for data processing to be achieved low latency and high bandwidth as well as security and fault-tolerante for sensor application close to the edge computing with real-time demands. as shown in [42], [43]. Fog computing architecture should be preferred when it is difficult to find stable power sources at the edge.

Cloud Computing In cloud computation platform, data sources are processed and fused in the cloud to provide resources in a transparent and usable form, where the application can adapt the resource to its requirement, This is the most common technique practiced by industry and research institutes for processing big data. Examples of this architecture being used are [44], [45]. The advantages of cloud computing architecture includes ready access to the data and both online and offline for further processing or fusing. The disadvantages include increased communication overheads and costs, large latency between data capture and computation, and excessive ingress bandwidth consumption.

Hybrid Computing In hybrid computation platform, processing is distributed among two or more layers in edge, fog and cloud as shown in [46], [47]. In this architecture, depending on the available resources or application objectives, some low level data fusion and processing is done at the edge or fog, while high level information is extracted in the cloud. The use a hybrid approach is the ideal way to process sensor data since all advantages and levels sensor data processing techniques are employed.

3.2.1.2 IoT Technologies

Even though data fusion in IoT paradigm is somehow new, there are various technologies, protocols and frameworks that target it. For such we can cite ZigBee, 6LoWPAN, BLE, IPv6, CoAP, MQTT, SOA, etc as known technologies and protocols used in Internet of things to enhance,process, and evaluate the performance of product. [15],[7],[14].

3.2.2 Description of some Solutions

3.2.2.1 ECVID

[48] proposes an Edge Computing architecture that provides an intermediate computing layer for IoT data. The proposed architecture uses VID (Virtual IoT Devices) for local data processing, management of physical IoT devices and quick reaction using actuators. The concept of VID is characterized as: (i) a virtualized instance of one or more sensors or actuators, (ii) hosted in a Cloud or Edge Computing platform and (iii) provides device description including a list of capabilities in terms of events, properties and action to facilitate data processing and communication to actuators. The Edge Computing system is running on a Raspberry Pi3 hardware and supports IoT devices exchanging data using HTTP and CoAP over BLE and WiFi. The platform is used in the context of a smart city scenario where connected vehicles provide sensor data about city temperature. Such architecture in turn reduce latency, improve QoS, allow real time data analysis and actuation resulting in superior user experience in consumer IoT applications and services.

3.2.2.2 EFCP

[49] proposed an edge and fog computing platform which facilitates the integration of complex heterogeneous sensors within a common management framework. The main contribution is the symbiotic hardware/software design approach, with the implementation of an efficient data fusion strategy. It is achieved by treating the sensors as services in a SOA and using a messaging abstraction that provides common functionality that can be tailored to each specific requirement in the communications path. Sensor services can identify themselves if they are real physical systems, or can be defined by a standard JSON representation which defines the operation of each virtual, or locally configured sensor. Regardless of their origin, sensors are defined by the messages they produce and consume making them a standard element within the framework. Therefore, the definition of a sensor provides an efficient and well-established interface, making it trivial to design a virtual sensor, providing a common ground for heterogeneous data sources. The SOA and abstract messaging system has been embedded in a modular hardware platform based on a layered architecture which provides a set of digital, analog and power-supplying interfaces. Heterogeneous physical sensors may be designed for each application in compliance with this hardware architecture, while the messaging structure allows their top-level integration into the overall system

by just defining the associated complex sensor messages. This approach could be especially beneficial for those applications where data provided from a set of complex sensors must be fused to obtain the desired output; for instance, think of a radar that provides distance information of the detected objects that has to be integrated to generate their exact geographical position.

3.2.2.3 OpenHab

The open Home Automation Bus (OpenHAB) is a platform for home automation applications. It provides the ability to connect a large number of devices and systems [50]. openHAB communicates electronically with devices in the smart home environment and performs user-defined actions. openHAB uses MQTT as M2M/IoT connectivity protocol to pushes the data to the public cloud for further processing. Various data (e.g, parameters of hue lights like color, brightness, and saturation) can be fused and controlled by REST API in order to create more accurate information making faultless actions in that smart home. openHAB is developed with Java so it works on Linux, Windows and MAC OS. OpenHAB is very flexible and customizable, and what's more important is open source. But it comes at a cost, you have to invest time to learn its concepts and set up an individual system adapted to your needs. Many parts of the installation require text-based configuration, potential access to log files for debugging, etc. The configuration of openHAB is therefore mainly reserved for technophiles. It is not a commercial product available on the market. In addition openHab does not support many types of devices which present a compatibility issue.

3.2.2.4 OpenIoT

[51] proposed OpenIoT, an open source IoT platform enabling the semantic interoperability of IoT services in the cloud. OpenIoT promotes interoperability among IoT silos right from the sensor to the cloud services. OpenIoT is built upon semantic web standards such as W3C Semantic Sensor Networks (SSN) ontology, which provides a common standards-based model for representing physical and virtual sensors, RDF to store, index and retrieve data, and supports virtually any IoT protocols such as CoAP, 6LoWPAN etc. OpenIoT includes also sensor middleware and sensor data fusion capability at the things and at the cloud. OpenIoT eases the collection of data from virtually any sensor, while at the same time ensuring they are embedded with proper semantic annotation. Furthermore, it offers a wide

range of Doit-yourself visual tools that enable the development and deployment of IoT services and applications with almost zero programming. Another key feature of OpenIoT is its support for mobile sensors and thereby enabling support for an emerging wave of mobile crowd sensing applications. The OpenIoT platform is a blueprint architecture to develop semantically interoperable smart city solutions with support for complex sensor data fusion algorithms.

3.2.2.5 SmartCity

Zanella et al., [52] provide an overview of the techniques, architecture and protocols available for an urban IoT system realized in the city of Padova, Italy. The application consists of a system for monitoring public lighting by means of wireless nodes, placed on streetlights and connected to the Internet via a gateway. Each IoT node is geographically located, so IoT data can be enhanced with context information. The nodes are equipped with photometric sensors which directly measure the intensity of the light emitted at regular time intervals or on demand. Wireless IoT nodes are also equipped with temperature and humidity sensors, which provide data on weather conditions, and a node is also equipped with a sensor which monitors air quality. data is collected, fused using IoT technologies. Smart City adopts IETF standards which are open and free. The IETF Standards for IoT provide a web service architecture for IoT services. This approach enables the development of flexible IoT services that can easily interact with other web services by adopting the REST paradigm. In particular, common standards for Internet communications, such as HTTP, IPv4, and Ethernet, are being superseded in resource-constrained devices (like sensor nodes) by their IoT counterparts, i.e., the constrained application protocol CoAP , IPv6 and 6LoWPAN. The protocol stack used in the solution is suitable for restricted nodes in terms of resources, the use of CoAP at the application level will considerably reduce the communications. The architecture of the solution based on the REST paradigm allows better scalability and flexibility of the solution. The solution does not use application-level communication encryption, although it could be beneficial for the lightness of the solution but the user data will not be sure.

3.3 Mathematical Data Fusion Methods for IoT

Some recent surveys give an overview of data fusion solutions in IoT environment. Lee et al. [53] provided a review on data fusion techniques, algorithms and theories, but they did not give a discussion on the data fusion specifically designed for IoT environment. Alam et al. [4] presented a critical review of data fusion for IoT with a particular focus on probabilistic, artificial intelligence, and theory of belief methods in specific ubiquitous environments but ignored the security issues in IoT. [54] proposed classification of data fusion methods into three categories: stage-based methods, feature-level-based methods, semantic-based methods. This based on data properties including imperfection, correlation, and insistences. However, Do not touch concrete IoT application scenarios. [55] investigated and classified data fusion methods according to challenging problems of input data, without discussion of application in IoT environment. Pires et al. [56] Focus on sensing and fusion methods for identification of activities in daily life, but ignores the privacy issues and the relationships among different applications in IoT. Wang et al. [39] reviewed and analyzed existing work according to an evaluation framework that consists of several features, which based on configuration, data processing, sensors, and portability. Ding et al. [33] specified data fusion requirements, pointed out the differences and characteristics of popular IoT application domains, and emphasized security and privacy issues in data fusion of IoT. However, they neglected the design of fusion methods in IoT environment. Though above surveys on data fusion in IoT, these are mainly focused on specific applications areas or classifications based on appointed features or security and privacy issues in the process of data fusion. In this section, our taxonomy and the different performance criteria used to evaluate the existing solutions are presented.

3.3.1 Classification and Taxonomy

As depicted in Fig. 3.2, a taxonomy of data fusion solutions the aim to process data in IoT environment is given. Data fusion solutions consider, besides energy consumption, data reliability, data accuracy and latency as critical metrics. Data fusion solutions are classified according to mathematical method into three categories: (i) measurement methods; (ii) characteristic methods; and (iii) decision methods. Further, each category is sub-classified based on its techniques and objectives. Some data fusion method aim to enhance data quality (data reliability, data accuracy), whereas the others aim to reduce data latency.

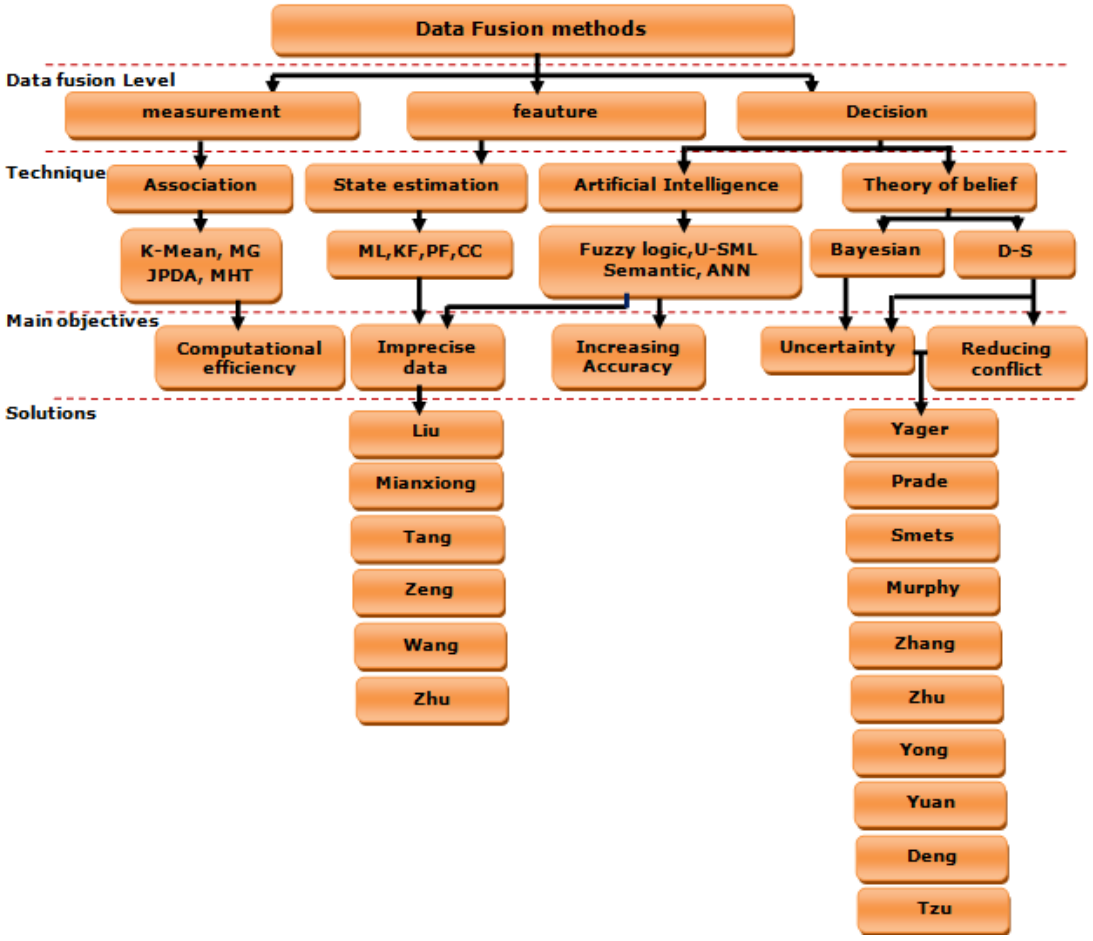


FIGURE 3.2: Taxonomy of mathematical data fusion methods in IoT

3.3.1.1 Association Technique

Data association considered as low level data fusion technique, the raw data measurements are directly provided as an input to the data fusion process, which provide more accurate data than the individual sources. data fusion is based on similarity between at least two or more data sources. Common techniques for data association include K.Means [57], Probabilistic Data Association [58], and Multiple Hypothesis Test [59], and Graph Model.

3.3.1.2 State Estimation Technique

State estimation is essential step in data fusion, which used mathematical method namely probabilistic and theory of belief in a specific IoT environment such as nonlinear and big data. The estimation problem involves finding the values of the vector state (e.g., position, velocity, and size) that fits as much as possible with the observed data. SBI (Sequential Bayesian Inference) is a method of analysis that

combines information collected from experimental data with the knowledge one has prior to performing the experiment. Common techniques under this category are ML (Maximum Likelihood) [60], KF (Kalman Filter) [61], PF (Particle Filter) [62], and CC (Covariance Consistency Model) [63].

3.3.1.3 Decision Technique

In this kind of technique, A decision is typically taken based on the knowledge of the perceived situation, which is provided by many sources in the data fusion domain. These techniques aim to make a high-level inference about the events and activities that are produced from the data sources. These techniques often use symbolic information, and the fusion process requires to reason while accounting for the uncertainties and constraints. Further, methods of decisional level can be divided into two subclasses:

Artificial Intelligence AI (Artificial Intelligence) can be defined as an area of computer science that seeks to build intelligent machines which can think or work like humans. In data fusion, AI enables the actuators to take highly accurate and informed decisions based on the sensed data. AI can play a crucial role in the IoT paradigm, especially in the areas where decision making and prediction are of vital importance. Developing intelligence is a gradual process, acquired by supervised/unsupervised machine learning [64], Artificial neural networks [65], Fuzzy Logic [66], and semantic approaches [67].

Belief Theory This type of decisional fusion methods has been proposed to combine evidence according to the probability theory rules, where uncertainty is represented using the conditional probability terms that describe beliefs. The belief theory provides a formalism that could be used to represent incomplete knowledge, updating beliefs, and a combination of evidence which allows to represent the uncertainty explicitly. Common techniques under this category are Bayesain inference [68] and D-S (Dempster-Shafer) inference [69]. Detailed description of this technique is presented later.

3.3.2 Evaluation Parameters and Performance Metrics

With respect to challenge of IoT environment including big data, Heterogeneous, privacy and imprecise. Different systems or solutions are evaluated across a variety of performance metrics, including accuracy, precision, computational complexity, robustness, and scalability.

3.3.2.1 Imprecise

Imprecise is an essential performance criterion. Data perceived by various sensors may be uncertain, inaccurate due to data loss or data source unreliability, which brings additional challenges for data fusion caused by data imperfection, data conflict, data ambiguity and inconsistency.

Data Imperfection: Collected data is sometimes inaccurate; it can be inaccurate and uncertain. This behavior is not infamous in wireless sensor networks. The imperfection must be dealt with effectively with the use of data fusion algorithms.

Ambiguities and Inconsistencies: Impreciseness is not the only factor responsible for data inconsistencies; the environment in which a sensor operate is largely responsible as well. Outlier detection, replacement and data imputation are vital in IoT environment.

Conflict: The conflicting nature of data can give rise to counter-intuitive results. The problem of conflicting data is visible more in evidential belief reasoning and Dempsters rule of combination. The data fusion algorithm must take critical care while treating conflicting data.

3.3.2.2 Energy Saving

Wireless Sensor Networks which is part of IoT environment consist of a large number of source nodes with limited capabilities, especially the strictly limited energy. in-network data processing, such as data fusion can significantly improve the energy-efficiency of the networks. Through this dissertation, the data fusion method will be discussed and evaluated in terms of energy consumption.

3.3.2.3 Computational Complexity

Computational complexity is provided by authors and includes reported details about algorithmic complexity or runtimes. Qualitative assessments such as low, moderate, and high may be provided when published information adequately supported such an evaluation.

3.3.2.4 Scalability

The implementation strategy should be scaled from a small simulation to a big scale. Thus, for an IoT environment, data fusion architecture scalability is of utmost importance. anothers parameters shoud be considered includes the size of the space in which experiments were conducted. Where possible to assess, and also the dimensions in which the approach was tested or to which it may extend.

3.3.2.5 Robustness

The comments of summarizes authors are based on robustness, inculding a qualitative assessment such as low, moderate, or high, if possible, of the solutions robustness to interference, noise, incomplete information, etc.

3.4 State Estimation Methods Based on Distributed Particle Filter

State estimation methods are indispensable in data processing, which used probabilistic and theory of belief techniques in a definite IoT environment such as nonlinear and target tracking. Target tracking is a dynamic state estimation problem prevalent in several area, including air traffic control, autonomous vehicles and robotics, remote sensing, surveillance, and computer vision [70] [71]. The problem basically concerns inferring the state of the target that is assumed as a random variable, by observing it or another random variable associated with it, namely observations. SBI (Sequential Bayesian inference) is a method of analysis that combines information collected from experimental data with the knowledge one has prior to performing the experiment. In particular, the particle filter is one of the most vital tools for realizing SBI [72], which uses a set of weighted samples

(called particles) to approximate the Bayesian prior and posterior, also known as SMC (Sequential Monte Carlo) [4] [23]. In recent years, particle filter has been successfully applied into many fields such as robotics, data processing. The advantages of the PF is that is not restricted by the linear and Gaussian assumptions [73], which makes it applicable in a wide range of wireless sensor networks and hence to the IoT applications. Also PF does not make any assumption on the measurement noise distribution. However, the estimation accuracy of the PF can also be degraded due to the range measurement uncertainty, also there are expensive computational demands in this approach.

To develop a distributed particle filter, there are two most important questions need to be handled. The first one is that what information we should communicate between sensors. The second one is that how we fuse the information contained in each sensor.

3.4.1 Distributed Particle Filter

Particle filtering methods are Bayesian methods that recursively approximate the posterior distribution of the unknown kinematic parameters using a discrete measure with a random support. The discrete measure is defined using a set of samples, also refereed particles, and their associated weights. The particles are sequentially drawn from an importance function through Monte Carlo technique, and the weights are recursively computed using the prior and the likelihood function of the kinematic parameters [74].

Particle filtering methods can be implemented in a distributed manner on sensor networks. In such implementation, each of the sensors on the network locally runs a particle filtering algorithm and exchanges information with the other sensors to approximate the global posterior distribution of the kinematic parameters [75]. However, after a number of measurements, most particles have negligible weight. It is the phenomenon of degeneration of weight. The particle system is depleted and therefore can no longer correctly represent the density of probability with the consequence of a possible divergence of the filter. To correct this phenomenon, a step additional so-called resampling is introduced. Higher weight particles are favored by replicating them identically, those of low weight which are found in the least probable regions are little or not at all chosen and disappear. This step allows to concentrate the capacity of the network. the state estimation model is applied to describe the nonlinear estimation problem based on the Markov process.

require two models, namely, state model and measurement model. The former describes the evolution of the state with time while the latter defines the relationship between the noisy observations and state. A state system model for the wireless sensor networks is considered as follows [71].

1) The state transition model: $x_n = Fx_{n-1} + Gu_n$ where x_n is a vector that denotes unknown states of the dynamic system at time n, F denotes the function that describes the time-evolution of the vector x_n and G is a vector of the state noise.

2) The measurement model: $Z_{k,n} = h_k(x_n) + e_{k,n}$ where $Z_{k,n}$ is a vector that describes the measurement obtained from sensor k at time instant n, and h_k the measurement function that maps the kinematic vector parameter x_n to the measurement vector $Z_{k,n}$.

The particle filter algorithm is presented in Algorithm. 3.1 as follows:

Algorithm 3.1: particle filter algorithm

Prediction :$\widehat{x_n} = f_n(x_{n-1})$
prior Measurement: $\widehat{z}_n = h(\widehat{x}_n)$
//Importance sampling
Draw $\left\{x_n^i\right\}_{i=1}^{Ns} \sim \mathcal{N}(\mu, \sigma)$
//Measurement
For particle i=1: Ns do
Compute likelihood function: $\left\{x_n^i\right\}_{i=1}^{Ns} \propto p(x_n^i|x_{n-1}^i)$ state transition model
Weight: $\left\{w_n^i\right\}_{i=1}^{Ns}$ $w_n^i \propto p(z_n^i|x_n^i)$ observation model
End For
Normalizing: $w_n^i = \frac{w_n^i}{\sum_{i=1}^{Ns} w_n^i}$
Resampling: $\left\{x_n^i, , w_n^i\right\}_{i=1}^{Ns}$

3.4.2 Kullback-Leibler Divergence

The KLD (Kullback-leiber Divergence), named also the relative entropy, computes the divergence between two pdfs (probability density functions). In the case of gaussian distributions, it evaluates the Burg matrix divergence between covariance matrices used in some optimization problems [76]. The KLD is a non-symmetric

measure of the difference between two probability distributions p and q ; p represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution. The measure q represents a theory, model, description, or approximation of p. KLD is defined in Eq. 3.1 as follows :

$$K(p,q) = \sum_{x} p(x) log \frac{p(x)}{q(x)}. \tag{3.1}$$

It represents the average of the logarithmic difference between the probabilities p and q, where the average is taken using the probabilities p.

3.4.3 Literature Overview and Discussion

In this sub-section we present a state-of-the-art of improved distributed particle filter solutions focused on single target tracking. Contains those that solve the problem of sample impoverishment and weight degeneracy [77], computational efficiency systems (estimation accuracy) [78] and importance sampling proposal [74]. Existing distributed particle filter solutions can be classified into two main categories. One is to adapt the proposal using an approximation of the Kullback-Leibler Divergence in order to avoid degeneracy, while the other is to move the particle cloud through the sequence of densities to reduce weight degeneracy.

In the first group, a recent attempt is given by automatically adapting the proposal using an approximation of the KLD between the true posterior and the proposal distribution, based on optimal adjusted variance and gradient data [[79], [80], [81]]. This is by increasing variance inversely proportional to the likelihood and creating new samples near the true distribution or the high likelihood region. More systematically, (group/layered/heretical) multiple importance sampling schemes use a set of different proposal distributions for better robustness [[82],[83]], by ensuring that an appropriate proposal density is obtained automatically. These, however, come at the expense of a moderate increase in the complexity.

In the second group, representative efforts include progressive consensus-based particle filter algorithms. Consensus means a global agreement on some quantity that depends on the data of all sensors, and a consensus algorithm specifies the corresponding information exchange between neighboring sensors and the computations performed locally by each sensor. Consensus algorithms are iterative schemes that diffuse information through the network and, usually, reach a global agreement

only asymptotically. There is no single point of failure, and the algorithms are robust to changing network topologies and unreliable network conditions such as link failures. To reduce the number of particles, and hence the communication requirements, the DPF presented in [71] uses an average consensus filter or forward-backward propagation strategy that limits the information exchange of each sensor to only its neighbors. In [84], a distributed implementation of an auxiliary PF is proposed. For distributed weight calculation, a modification of the randomized gossip algorithm known as selective gossip algorithm [85] is used. Here, the communication requirements are reduced by transmitting only information about the largest weights. In [86], two alternative algorithms for distributed weight computationbroadcast gossip and belief propagation are investigated and compared regarding their performance and convergence speed.

In Chapter 5, We will present a new method to choose the appropriate sample used in particle filter algorithm, this by comparing two probability density functions. The proposed solution used a new method to measure the similarity distance between functions, such measure optimize the number of particles used in resampling phase, and hence calculate the weights with more precision.

3.5 Dempster-Shafer Data Fusion Methods

The Dempster-Shafer theory [87] is largely used for uncertainty reasoning, which allows processing uncertain or imprecise information without prior knowledge. This can be well used in IoT environment. It supports the representation of both imprecision and uncertainty, and it allows deriving the probabilities of a collection of hypothesis while dealing with missed information. This can be helpful to process the heterogeneous IoT data. However, under situations where the evidence highly conflicts, it may obtain counterintuitive results.

Several protocols based on D-S theory have been proposed in IoT environment that aims at improving decision-making. Some of these protocols rely on the combination rule, while some others are interested in the measurement of distance between nodes (weight) or amount of information. The relevant literature is reviewed in this section based on different categories of D-S approaches. We propose a taxonomy of the state-of-the-art protocols that are based on D-S theory using several criteria as shown in Fig. 3.3, and then we describe the most common methods. Two main categories may be distinguished, 1) methods based on modified models, and 2) improved D-S based on modified methods.

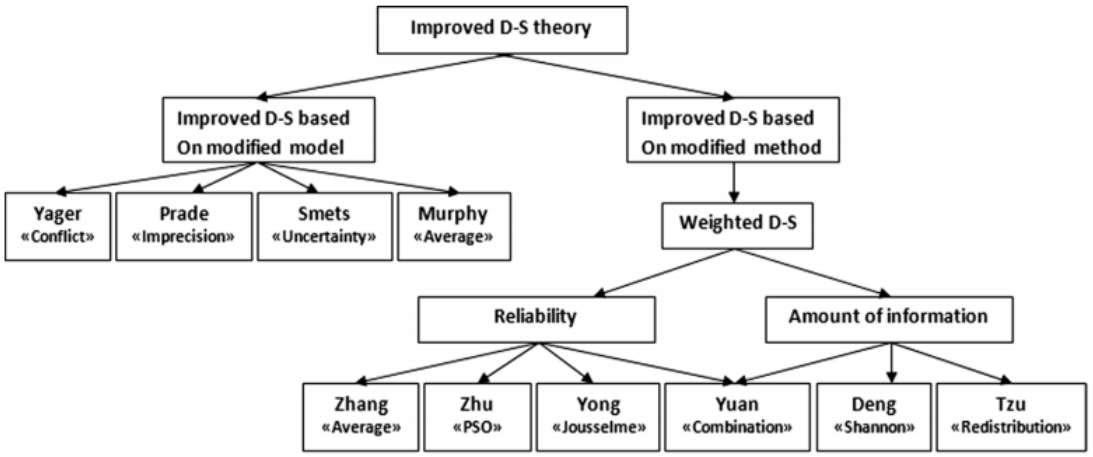


FIGURE 3.3: Taxonomy of D-S based data fusion methods

3.5.1 Principle and Classification

The basic concept of the D-S evidence theory was originally developed in (1967) [88], and then evolved towards belief functions to model uncertain knowledge on the basis of mathematical formulations [89]. In D-S reasoning system, possible assumptions consist of all elements of indivisible hypothesis that are mutually exclusive and comprehensive. This includes a frame of discernment, denoted T, the space of inference system, θ, which includes all possible subsets of T elements. The number of possible combination (including the empty set) is 2^n, where n is the number of elements in T. There are three main functions in D-S theory, i) the basic belief mass function that specifies the belief mass distribution (m-values) over all possible sub-sets of a frame of discernment, ii) the Belief function, and iii) the Plausibility function.

The Mass function, which is also called a basic probability assignment, associates for every element, $E \in \theta$, $m(E)$ that is the proportion to all available evidence. The value of $m(E)$ therefore concerns only the state E and brings no credit to the subsets of E, which by definition have their own mass. The mass of the empty set is null ($m(\emptyset) = 0$), and the masses of θ's subsets sum up to 1, i.e., $\sum_{E\in\theta} m = 1$. The belief of a set, A, to a node, i, say $Belief_i(A)$, is defined as the sum of the masses of all A's subsets (Eq. 3.2).

$$Belief_i(A) = \sum_{E_k \subseteq A} m_i(E_k). \tag{3.2}$$

The plausibility measures the intensity with which the element, A, is found (with no doubt). It can also be interpreted as the maximum belief in A, or the sum of

the evidence that is not against A. $Plausibility(A)$ is defined as the sum of the masses of all sets that intersects with A (Eq. 3.3).

$$Plausibility_i(A) = \sum_{E_k \cap A \neq \emptyset} m_i(E_k). \tag{3.3}$$

The procedure for fusing multiple evidence using the previously defined mass functions is an important issue in the D-S theory, which provides a method to compute the orthogonal sum ($m = m_1 \oplus m_2$) of two bodies of evidence according to Dempster's combination rule [90].

When there are multiple sources and the observations are assumed to be independent of each other, the combination of evidence in D-S provides away to combine these observations. For every proposition A in, θ, the combination rule between an object with a mass, m_i, and another object with a mass, m_j, is given by Eq. 3.4,

$$m(A) = \frac{\sum_{B \cap C = A} m_i(B) m_j(C)}{1 - K}, \tag{3.4}$$

where K is a measure of conflict between the sources, which is also called inconsistency of the merger. It is given by Eq. 3.5.

$$K = \sum_{B \cap C = \emptyset} m_i(B) m_j(C). \tag{3.5}$$

The mass function obtained after combination tends to reinforce the belief of decisions for which the sources are consistent. Note that this combination of evidence rule is both associative and commutative. That is, the mass function can be the result of a combination of evidence between two other objects. The evidence of combination process for multiple sources can be chained, and the order in which the sources are combined does not affect the final results. The combination of conflicting evidence has always been challenging in D-S evidence theory [91]. Many methods have been proposed to solve this problem [90, 92, 93], but there is no universal solution thus far.

As depicted in Fig. 3.3, we classify the existing solutions of literature into two classes: (i) protocols based on modified models in which measure of conflict is

modified in combined rules and (ii) improved DS based on modified methods in which a weight coefficient is used in the calculation of BPA. In the remaining of this section, existing solutions belonging to each class are presented, their advantages and shortcomings are highlighted.

3.5.2 Description of Modified Models

Solutions of this category aim to reduce the conflict in combined rule by exploring the advantage of using D-S theory and mass function.

Yager [94, 95] suggested that all conflicting evidence is unable to provide effective information, so he assigned all conflicts to unknowns $m(\theta)$. The improved formula can be used in high conflicting evidence combination, but the irrational distribution will lead to unreasonable results for assigning all conflicting evidence to the unknown.

Smets [96] considers that the data sources are reliable. Based on this assumption, the conflict can only come from an ill-posed problem, i.e., the non-inclusion of one or several assumptions in the frame of discernment. Therefore, the author recommends redistributing the conflict mass K, but only on the empty set.

Dubois and **Prade** [97] consider that the data sources are the unreliable part and assume that when a conflict exists between two data sources, at least one of the two sources is reliable. Given the impossibility of identifying the reliable source, they opted for redistributing the conflicting mass on the union of the two sources. Dubois and Prades rule doesnt work for dynamic fusion problems when a singleton or a union of singletons becomes empty. This problem is fixed by the sum S2 in the general D-S rule of combination.

Murphy [98] averaging approach suggested that if all the evidence are available at the same time, the average of evidence masses is calculated and then combined N time using D-S theory, where N is the Number total of evidence. However, this approach does not consider the association relationship and difference among the evidence.

3.5.3 Description of Modified Methods

The second category which focused on modified method when applying D-S theory. This by using a weight coefficient to calculate the basic probability assignment. the modified methods are divided into two classes, reliability methods vs. the

amount of information based methods.

In reliability methods, the evidence distance of **Jousselme** [99] is used. The distance between two bodies of evidence, $d_{BOE}(m_1, m_2)$, is defined in Eq. 3.6.

$$d_{BOE}(m_1, m_2) = \sqrt{\frac{1}{2}(\overrightarrow{m_1} - \overrightarrow{m_2})^T D(\overrightarrow{m_1} - \overrightarrow{m_2})}, \tag{3.6}$$

where $\overrightarrow{m_1}$ and $\overrightarrow{m_2}$ are the vector forms of the evidence bodies. The size of each body is 2^θ. D is a $2^\theta * 2^\theta$ matrix, whose elements are given by Eq. 3.7,

$$D(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|}, \quad s_1, s_2 \in 2^\theta. \tag{3.7}$$

Yong [100] applied the evidence distance to obtain a weighted average combination and thus measure the conflict degree among evidence. The higher the distance between two bodies of evidence is, the less these two bodies of evidence support each other. If evidence conflicts highly with others, it will have less effect on the final combination result. The support degree and the credibility degree of each evidence are defined, respectively, with Eq. 3.8 and Eq. 3.9.

$$Sup(m_i) = \sum_{j=1, j\neq i}^{N} (1 - d(m_i, m_j)). \tag{3.8}$$

$$Crd_i = \frac{Sup(m_i)}{\sum_{j=1}^{k} Sup(m_j)}. \tag{3.9}$$

The credibility degree represents how reliable evidence is. The higher the credibility degree is, the more effective the evidence will have on the final combination result.

Zhang [101] proposed a new method of combining conflicting evidence based on average. This method considered the association relationship among the evidence collected from multi-sources, and it weighs the evidence based on the distance of evidence.

Zhu [102] proposed a new method for weighting evidence using PSO (Particle Swarm Optimization) algorithm to optimize the calculation of sources weight.

Context-aware data fusion [103–105] is employed in the basic concepts of IoT. The amount of information based methods use efficient tools to quantify information. These approaches can be applied in evidence theory where the uncertain information is represented by BPA.

Deng entropy [106] is one of these methods and a generalization of Shannon entropy [107]. It is an efficient way to measure uncertainty, not only under the situation where the uncertainty is represented by a probability distribution, but also in the situation where the uncertainty is represented by BPA. This enabled its wide application in D-S evidence theory. When the uncertainty is expressed in the form of a probability distribution, Deng entropy degenerates to Shannon entropy.

The related concepts are given in the following. Let A_i be a proposition of BPA m; the cardinality of the set A_i is denoted by $|A_i|$. Deng entropy E_d of the set A_i is defined by Eq. 3.10

$$E_d = -\sum_i m(A_i) log \frac{m(A_i)}{2^{|A_i|} - 1}. \tag{3.10}$$

Tzu [108] proposes another method to calculate information volume, which reapportions the mass in the null set among the other subsets that are originally assigned to the null set. This is using Eq. 3.11.

$$I_{ev} = \sum_{i=1, A_i \neq \emptyset}^{n(A_i)} \frac{m(A_i)}{|A_i|}. \tag{3.11}$$

Yuan [109] combines the weighted credibility method in [100] to reduce conflict between evidence, and Deng entropy when calculating information volume.

Contrary to most of the previous methods that are pointed to only one parameter to calculate the weighted evidence, the one proposed in this article improves the credibility degree by considering the evidence relationships, and enhancing the uncertainty degree using inner properties of evidence. We are particularly interested in the reliability of a sensor which is determined by the distance between the sensor and the entity in question, as well as the time validity of the information (lifetime).

3.6 Conclusion

This chapter investigated the data fusion principle and focused on features and requirements that should be implemented by data fusion methods as well as platforms. In particular, the chapter reviewed existing approaches by proposing a taxonomy to classify frameworks data fusion in IoT environment. Based on our study in this chapter, choosing the best method to fuse data with dynamic, real time data processing and scalability issues. The surveyed mathematical solutions for data fusion aim to reduce the data latency, energy consumption, increase the data accuracy and ensure data reliability. We have proposed a clear taxonomy to classify existing solutions into three main levels: measurement, feature, and decision. more focused on state estimation solutions, Existing distributed particle filter solutions can be classified into two main categories. One is to adapt the proposal using an approximation of the KLD in order to avoid degeneracy, while the other is to move the particle cloud through the sequence of densities to reduce weight degeneracy. The chapter also reviewed decisional methods based on D-S theory. We have classified existing solutions into two classes: modified methods in which measure of conflict is modified in combined rules; whereas the second one is modified methods in which a weight coefficient is used in the calculation of BPA. Unlike the modified models methods that not consider the association relationship and difference among the evidence in data fusion, modified methods-based solutions regard the credibility degree by considering the evidence relationships, and the uncertainty degree using inner properties of evidence. For this reason, the weighted D-S solutions has been more attractive than the modified model-based ones in recent years.

Chapter 4

Service-Based Frameworks for Data Processing in IoT

4.1 Introduction

IoT data processing based solutions leverage advances in web and mobile applications, especially with the emergence of service oriented architecture, to provide flexible, dynamic, adaptable, and above all, easy to use administrative and control applications [7]. Software applications will be installed on mobile devices (smart phones, tablets, PCs, etc.) which will considerably increase the scope of the solutions. However, despite the wide efforts in web technologies adoption and standardization for the IoT, designing a scalable and extensible IoT framework that meets IoT functional requirements remains challenging. This has been considered in this work, in which we designed and implemented data processing frameworks to manage and control web/mobile applications in IoT using IPv4/IPv6 and higher layer protocols such as CoAP, HTTP, WebSocket. The frameworks allow anomaly detection in IoT devices and real-time error reporting mechanisms.

In this chapter, we propose two efficient data processing frameworks to ensure real-time data processing and reduce the network energy. The first solution is based on edge computing which integrates sensors and RFID technologies to enable sophisticated services via the Internet. In contrast to the existing approaches, and by processing data at the edge of the network, communication overheads can be significantly reduced and hence reduce energy consumption. The second solution is an efficient hybrid computing framework for data management and control

in smart cities to fuse heterogeneity data. The framework allows heterogeneous resources, connectivity, reliability and mobility. It ensures security and contains services to enhance data processing in application framework architecture.

The rest of this chapter is organized as follows. The framework based on edge computing is presented in Section 4.2. The proposed approach process data at the edge of the network via several services. In Section 4.3, we present our second solution, efficient hybrid computing framework for data management and control, which combines heterogeneity data. The process is tackled in edge, fog, and cloud computing. As an instantiation of the proposed frameworks, we consider the smart parking and smart home scenarios for the implementation and tests. We thus implemented the different modules via an IoT enabled mobile application, and made extensive tests. The results show considerable reduction in cost and energy consumption. We draw conclusions in Section 4.4.

4.2 Framework Based on Edge Computing

4.2.1 Framework Description

As seen in Chapter 3, the main objectives of IoT data processing frameworks are to reduce the computation and storage cost, improve network transmission reliability, reduce the network delay, enhance IoT network security, and ensure scalability. We propose an application dependant framework that we implemented for car parking management in smart cities as a typical application. The main contribution is the symbiotic hardware/software design approach, with the implementation of data processing strategy. The integration of networked sensor/actuator and RFID technologies enable sophisticated services in the emerging Internet of things context. The framework uses an active RFID tag per vehicle. The tag can be allocated to a subscribed customers over a long period of time, or it can be dynamically provided to the transient customers at the entrance. Each parking lot is equipped with a sensor (ultrasonic sensor are used in the implemented prototype) that is connected to a wireless mote. The mote will manage a bunch of sensors connected in serial mode. Data are processed in the edge to preserve data privacy and to save the bandwidth for relaying data. With this architecture, communication overheads can be significantly reduced as well as energy consumption. Additional, the service is proposed to customers via mobile device application. It is achieved by treating the sensors as services in a SOA and using a messaging abstraction that

provides common functionality that can be tailored to each specific requirement in the communications path. Sensor services can be defined by a standard JSON representation which defines the operation of each configured sensor.

4.2.2 Framework Architecture

The proposed framework builds upon a four-layer ubiquitous architecture: i) A sensor layer hosts the hybrid sensors and is used for parking spot detection and security. ii) A network layer is layered above the sensor layer to enable dissemination of the information between sensors and gateways. It uses both wired and wireless communication. Sitting on top of the network layer, iii) a middleware storing data and enabling visualization is used as an interface between the network layer and, iv) the application layer, where different services related to the smart parking are implemented. The implementation of the Framework includes three parts that are connected using a LAN network (WLAN or Ethernet). i) The first part is the **parking manager**, where all data are stored. It may provide some information for clients over internet, e.g. parking spots availability. The payment service is also provided by the parking manager within a park. ii) The second part is the **gate manager** with main function to control the gate using WSN and forward the gate status to the Parking Manager. iii) Finally, the **parking spots manager** is responsible for monitoring the parking spots and the cars within the level. WSN is used within the level to carry the status of spots and cars to the sink in a level as depicted in Fig. 4.1. This would be transmitted using a LAN network. The proposed framework scales to a multi-level car parks.

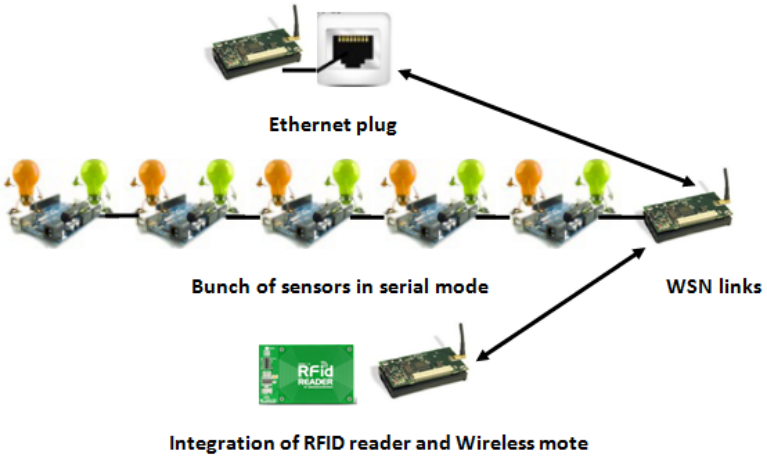


FIGURE 4.1: Framework architecture

Existing data at the sensor mote (Master) need to be forwarded to the sink of a level using multi-hop communications. For this purpose, we propose to use LIBP [110] data gathering protocol that provides efficient management of energy consumption in heterogeneous wireless sensor networks. In WSN with multiple types of motes having different levels of consumption, the aim is to ensure the network traffic can be managed to achieve balanced lifetime for all the motes in the network. This is a typical utilization in the car parking management, featured by the co-existence of: i) simple node as guiding node with moderate energy consumption, ii) the master node responsible for parking lots management, which consumes more power, and iii) the hybrid node (integrating the wireless mote and RFID reader) that is the most power consuming.

4.2.3 IoT Data Processing in Edge Computing

The flexibility of the integration of WSN and RFID network enables to provide many services that facilitate data processing. Here the data concerns the identity of the car (tags ID) and the location of the car in the parking lot. The following are some examples of these services:

4.2.3.1 Car Retrieval Service

A common problem for clients is when they forget where they park their cars. The proposed framework provides a service that assists in retrieving a forgotten car location using the integration of WSN and RFID in the hybrid node, and an active RFID tag to be kept by customers. When a customer requests his cars spot using a trigger in the active tag, a hybrid node gets the tags ID and transmits it to the parking manager. This latter checks the occupied parking spots in the database and in the parking field using the WSN. The response is returned to the appropriate guiding node for display in the variable message screen.

4.2.3.2 Parking Spot Reservation

Bunch of sensors can be enhanced with a source of light in each parking spot. These lights are controllable by the wireless mote. They can provide information about the status of a spot, e.g., red for occupied, green for empty, yellow for reserved, and blue for out-of-service. the service of checking the free parking spot is proposed using REST API.

4.2.3.3 Gate Management Service

Another use of RFID tags is gate management. As example, a gate can be opened automatically using an RFID reader and the vehicles tag at the gate.

4.2.3.4 Availability Checking over Internet

The car parking framework provides a real-time availability checking over internet. Cars can be checked if they are in spots and drivers if they are in a car parking using their tags. REST has been chosen for its stateless architecture that generally runs over HTTP.

4.2.3.5 Parking Management Applications

Advanced applications for both computer (website) and smart phones (android) can be designed based on the proposed framework.

4.2.4 Implementation

The proposed framework has been implemented and tested in lab prototype as presented in Fig. 4.2. Client devices have been connected via TCP/IP protocol to a parking database. The latter is updated in real time with the status of parking lots. Two kinds of client applications have been considered for parking lots monitoring: 1) mobile device application, for phones and tablets, and 2) desktop application for laptops and desktop computers.

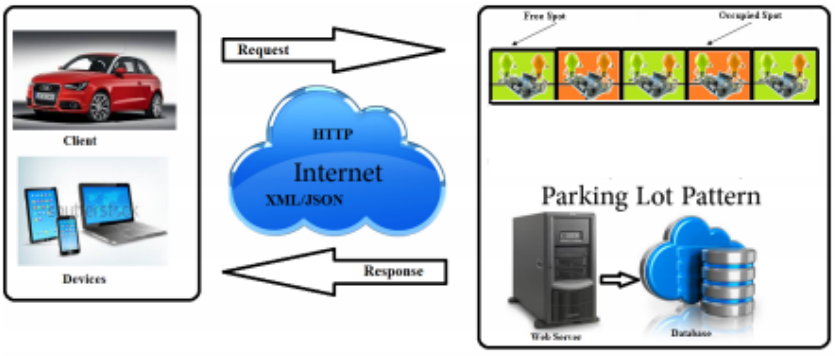


FIGURE 4.2: An overview of the prototype

In the prototype implementation, ultrasonic sensors are connected in serial mode using micro-controllers. All data are first transmitted to a master mote, then forwarded to the based station. This latter stores data in MySQL database for the client application. *I2C* protocol has been used to transmit data from the parking lots (micro-controllers with ultrasonic sensor) to the master wireless mote. In the prototype, ARM-based development boards have been integrated to the level gateway. Further, the proposed portable and efficient design for lot manager pattern has been implemented. In order to test the efficiency of the prototype, real-world experimentation on a bunch of lots has been carried out in our campus at CERIST research center. We have used for this experiment: i) Two wireless motes, a transmitter and a receiver. The transmitter is the sensor bunch master, and the receiver is for connection to the gateway, i.e., it is plugged to the gateway. ii) Three micro-controllers equipped with ultrasonic sensors, one is used per lot. iii) A PandaBoard: an ARM based board, as a gateway. iv) LCD Screen. When a car is parked in the free lot, it will be detected by the mote through the appropriate sensor. The data will be transmitted from the wireless mote (transmitter) to the receiver mote that is connected to Pandaboard. The event will be stored in the gateway of the level, which is in our case a Pandaboard ARM based computer. The Pandaboard provides the service of checking the free parking spot using JSON web server over the local network using WiFi or Ethernet.

We present a user application for car park management as shown in Fig. 4.3. It enables to obtain the number of free spot using mobile phones or tablet that are equipped with Internet connection. The user application accesses the database using web services and has two modules: i) smart parking module, ii) web site administrator for the management. Different technologies have been used. Android as the underlying operating system, and Java as the programming language with the android integrated virtual machine. The use of Java language enables developers to take advantage of Android library set up by Google, in addition to the standard Java libraries. Further, the interface of applications is constructed using an XML file, which allows for an SDK (Software Development Toolkit) that provides a graphical help interface for their construction. The paradigm of web services is based on a components architecture that uses Internet protocols to manage the communication between components, interact with a database, make the link between an external application and the database, provide access to content while keeping it safe. REST API has been chosen for its stateless architecture that generally runs over HTTP. REST involves reading a designated web page

that contains an XML file. The XML file describes and includes the desired content. Database server, MySQL, is defined as a management system database. A set of APIs for the development of business-oriented applications is used. The *J2EE* architecture is based on the Java language that allows the deployment of components on various platforms, independently of the programming language. We used web server Apache Tomcat, which is a server provided by JBuilderX to compile and execute the APIs (Servlets and Java Server Pages).

FIGURE 4.3: User interface

4.2.5 Performance Evaluation

We conducted a number of experiments and simulations to quantitatively investigate the impact of the proposed framework in terms of energy efficiency. The performance metrics include the energy consumption of a bunch of lots for locating parking spots and the energy consumption for data gathering from the sensors to the base station. We compared the proposed approach of using hybrid wired/wireless communication (integration in a bunch of nodes) with the standard approach of using a wireless mote per spot, as in [111], [112], [113]. Empirical experimentation using real equipment has been conducted for this comparison. For practical limitations, limited number of nodes are used in the experiments, but it is obvious that the difference will increase with the number of nodes. For each number of nodes (from one to six), the two approaches are compared. Fig. 4.4 depicts the cumulative energy consumption. It numerically demonstrates that the proposed solution is more engineering efficient than using one sensor per parking spot.

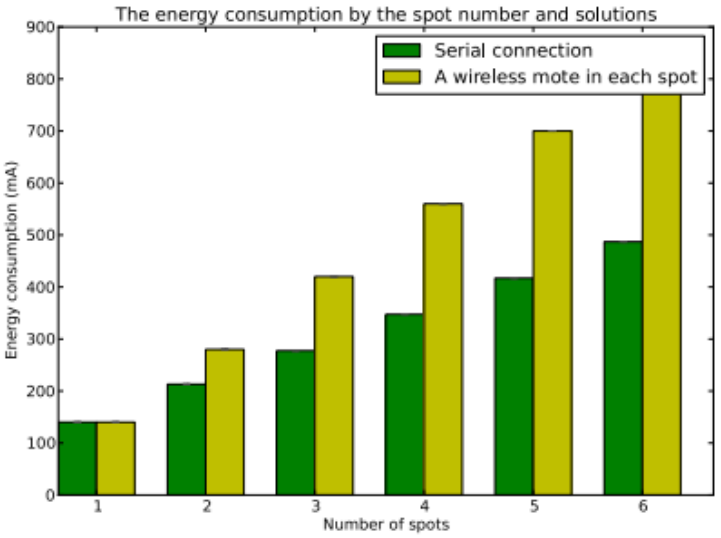


FIGURE 4.4: Energy efficiency

However, the proposed framework has limitations. First, processing data in the edge has a limited storage and knowledge about the environment (local context). Therefore, data processing cannot make high level decision where overall knowledge is required. Second, the proposed framework do not take into consideration the heterogeneity of data. For that, we propose in the following more generic framework to process heterogeneity of data based on edge, fog, and cloud computing.

4.3 Hybrid Framework

4.3.1 Framework Description

In this section, we present a design and implementation of a data management framework to monitor and control smart objects in IoT using edge, fog, and cloud computing. This is through IPv4/IPv6, and by combining heterogeneity data and also IoT specific features and protocols such as CoAP, HTTP and WebSocket. The framework allows anomaly detection in IoT devices and real-time error reporting mechanisms. Moreover, the framework is designed as a standalone application, which aims at extending cloud connectivity to the edge of the network with fog computing. Fog computing extends cloud computing to the edge of the network to eliminate the delay caused by transferring data to the remote cloud. It extensively uses the features and entities provided by the capillary networks with

a micro-services based architecture linked via a large set of REST APIs, which allows developing applications independently of the heterogeneous devices. The framework addresses the challenges in terms of connectivity, reliability, security and mobility of the Internet of Things through IPv6. The composition of each layer is illustrated in Fig. 4.5 and described in the following.

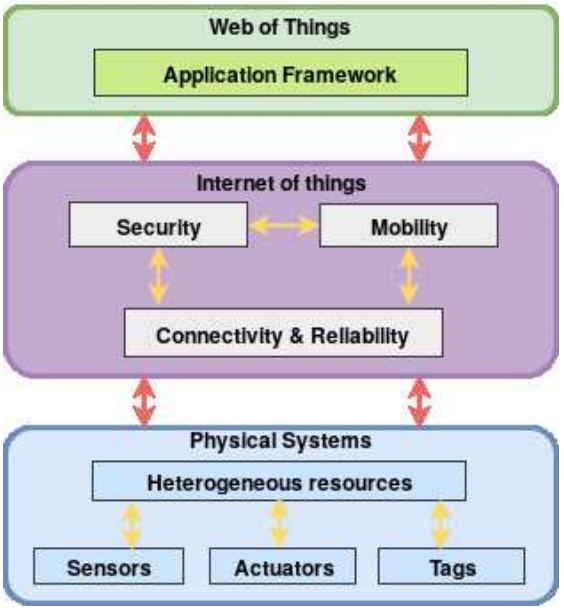


FIGURE 4.5: Framework description

4.3.1.1 Heterogeneous Resources

Since IoT ecosystems will be composed of a high range of technologies, a suitable support for the heterogeneity needs to be provided by the IoT communication architecture. In the proposed solution, we take into consideration sensors, actuators and tags. The use a RESTfull API to expose every object in its part boosts heterogeneity and facilitate the extensibility of the system. Furthermore, the wireless technologies exposed in our solution permit integration of different resources easily.

4.3.1.2 Connectivity/Reliability

The base of the Internet of Things is providing connectivity and reliability. IPv6 is the main enabler for extending the Internet of Things to the Future Internet. This work presents how the architecture has been powered by the IPv6 connectivity in order to provide an homogeneous, scalable, and interoperable medium for

integrating heterogeneous devices built on technologies such as 6LoWPAN, Bluetooth Low Energy (BLE). Using IPv6 allows each device to connect directly to the Internet and would allow billions of devices to connect and exchange information in a standardized way over the Internet.

As BLE doesn't natively communicate with IP, the best way to achieve this is to use 6LoWPAN. This simplifies the IP headers, compresses data and encapsulates the IP packets to allow them to be sent via Bluetooth efficiently, conserving bandwidth and power. The combination of BLE and IPv6 brings us much closer to the goal of having small, low-power devices that can communicate directly with each other and the Internet without using different hubs from each manufacturer sitting in the middle.

Nowadays, a variety of IEEE 802.11n wireless APs (Access-Points), including a dedicated commercial AP, a software AP, and a mobile router, can be used for wireless local-area networks. Along this trend, we use the Raspberry Pi as a Gateway that interconnects beacons and also as a device to run the software AP, because it provides a cost effective, energy saving, and portable embedded system. In this chapter, we present a conguration of Raspberry Pi for the software AP using IPv4/IPv6. We configure our software AP package which provides a script that combines hostapd package, which has WPA2 support, dnsmasq and iptables for the good functioning of the access point.

Since IPv6 infrastructure is not yet deployed in some countries, the communication to an IPv6 network is not possible through an IPv4 one, our solution surpass this constraints by offering a translation mechanism by the implementation of a CoAP-http proxy that translates the http-IPv4 requests to a CoAP-IPv6 requests and the same is done for responses.

4.3.1.3 Mobility

The use of an IPv6 based architecture will permit addressing large set of devices, and make them uniquely identifiable in the internet. Further, the adoption of a micro-services architecture will address more than one object using the same ip address and differentiate between them using the URI. This will make the communication with the objects independent of their location in the globe, which improves objects mobility. Additionally the exploitation of fog computing will essentially simplify direct communication with mobile devices and therefore enhance mobility.

4.3.1.4 Security

The security of our framework is ensured using Token-based authentication, DTLS protocol and passwords encryption.

Token-Based Authentication: Token-based authentication is a very powerful concept that provides a very high level of security. It is used to manage access to system resources by using a token without the need to send authentication data whenever a resource is requested. The sequence is as follows:

1. The client requests an access token for authentication by sending an HTTP request to the authentication server that contains the user name and password.
2. A response containing an access token and a refresh token is sent by the server if the received data is correct. An error message is sent otherwise. The sent token is unique and associated to a single user. It is characterized by an expiration date after which refreshment is necessary.
3. The client uses the token to request access to a resource. The token will be included in the header of each sent request.
4. Upon receipt of a request, the server checks the validity of the token. If it is not valid, an error message is returned.

The Fig. 4.6 shows the exchanges between the client and the server:

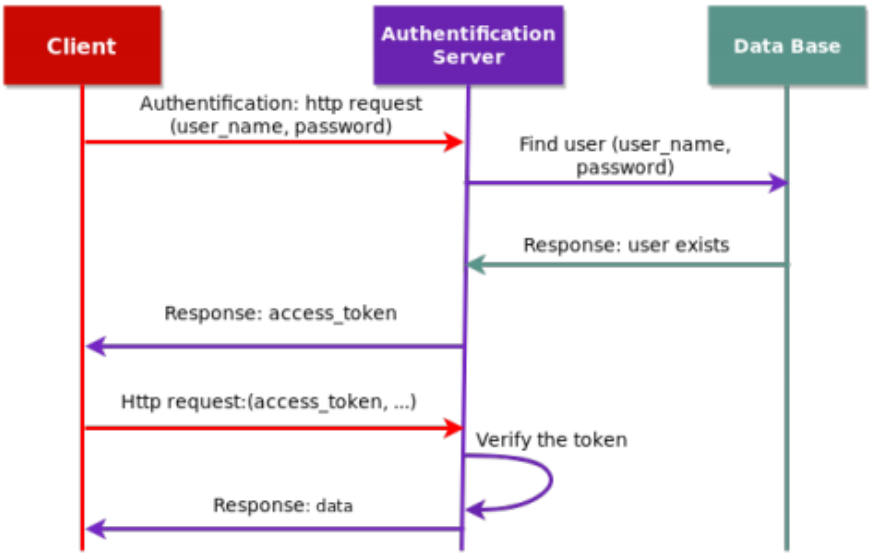


FIGURE 4.6: Token mechanism

DTLS: The DTLS protocol [114] provides communications privacy for datagram protocols. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. DTLS uses two algorithms Elliptic Curve Diffie-Hellman Exchange and Elliptic curve digital signature algorithm. In our solution, we implement DTLS in two parts: i) server: which contain implementation of DTLS in Contiki[115] using tinydtls[116] library. tinydtls has become an important tool for experimenting with DTLS in constrained devices by users from the academia as well as the industry, this part is used in the devices. ii) client : which contains implementation of scandium [117] and is used in mobile application.

Passwords Encryption: The user database is a critical element of the system, if a hacker gets access to it he will have access to all accounts of users, hence the importance of securing the users passwords. To ensure security, there are several hash functions such as Hash function with a fixed salt, Hash function with one salt per user and Slow hash functions (Bcrypt). After studying the different hash functions, we opted for the use of Bcrypt which gives us a high level of security. Bcrypt takes approximately 100 milliseconds to execute. It's fast enough so that the user does not notice it when connecting, but still slow enough to make it extremely expensive to run it to chop up a large list of frequent passwords. Bcrypt runs an internal encryption function multiple times, which slows down its execution. The number of loops is configurable which means that if we ever get

processors or GPUs 1000 times more powerful than the ones we have today, we can just re-configure the system and increase the number of loops. This will cancel the advantage of the new processors.

4.3.2 IoT Data Processing Architecture

The architecture on how IoT data is processed is proposed. The aims is to ensure efficient services in each level. The Iot data processing architecture using edge,fog and cloud computing is depicted as shown in the Fig. 4.7 bellow,

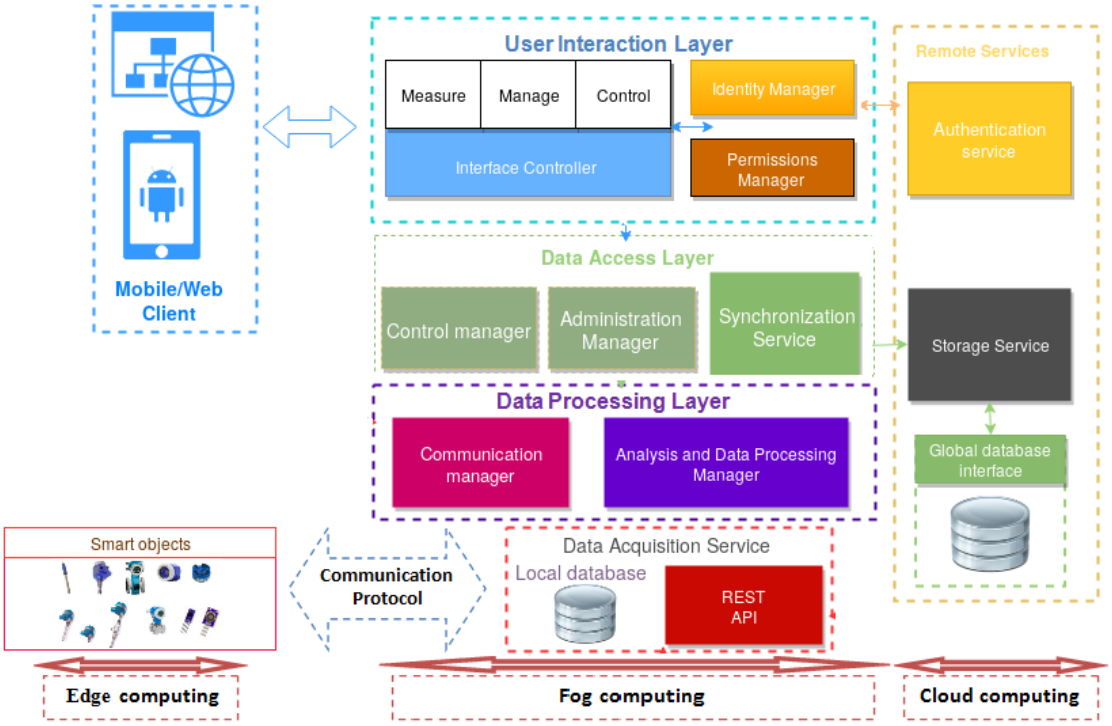


FIGURE 4.7: IoT data processing architecture

4.3.2.1 Data Acquisition

Objects intercept changes in the environment and communicate the data to the data acquisition service via the CoAP protocol. This service is also responsible for translating HTTP requests from the client to CoAP requests for querying objects. The local database will permit the principal of fog computing.

4.3.2.2 Data Management

It is responsible for the analysis of requests from the client, it deals with management of users and homes, and redirects the those concerning the control of objects to the acquisition service.

4.3.2.3 Data Access

It manages the different connections with the database (add, update, delete), the databases synchronization and the control of objects.

4.3.2.4 Data Encryption

To maintain privacy and security parameters, data obtained from the sensor are encrypted using DTLS protocol.

4.3.2.5 Data Presentation

It represents the interface of the system with the user, it includes the mobile interface built with Android and the interface of the website built with Angular.

4.3.2.6 Data Synchronization Service

A remote storage service is called upon to receive the data from local database (Raspberry Pi) to be archived in global database.

4.3.2.7 Authentication Service

A federated authentication service is used to allow the user to log in with their Google ID.

4.3.3 Implementation

There are a various application frameworks available for web and mobile development, each one has its specificities that make it usfull in some cases and not

in others. Because our solution targets an IoT environment, we opted to use a Javascript stack technologies in both frontend and backend because of the advantages offered by this language such as rapidity, asynchronous requests, interoperability with other languages, simplicity and extended functionality with third party add-ons. This stack is called MEAN Stack which refers to (MongoDB, ExpressJs, Angular, NodeJs). The web application will exploit the backend beside the frontend. Hereafter we explain in details each part and the technologies used:

4.3.3.1 Backend

It uses NodeJs, an open-source, cross-platform JavaScript run-time environment and exposes a Restful api with ExpressJS, to access data stored in a NoSql database managed by MongoDB, to fuse heterogeneity data for the purpose of the right decision (action) in a very short time. The communication with objects insured by an IoT CoAP application protocol with the implementation of Node-CoAP.

4.3.3.2 Frontend

It uses a component oriented framework which is angular based on typescript; a superset of JavaScript that allows to speed up development and facilitates code reuse. This uses stack of technologies called MEAN Stack as shown in Fig. 4.8.

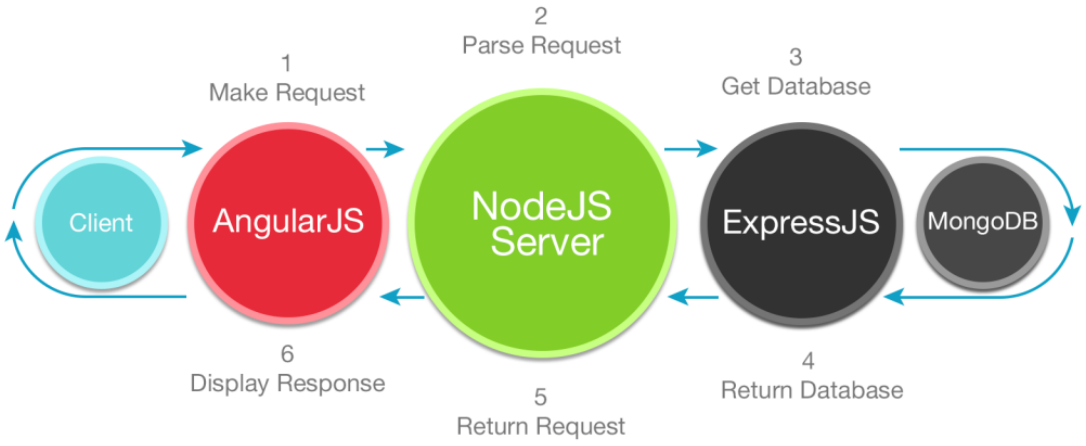


FIGURE 4.8: MEAN stack

The web application is used to control and administrate smart home, it allows to manage houses, including their rooms, users, and Raspberry servers with their sensors, actuators and cameras as shown in Fig. 4.9.

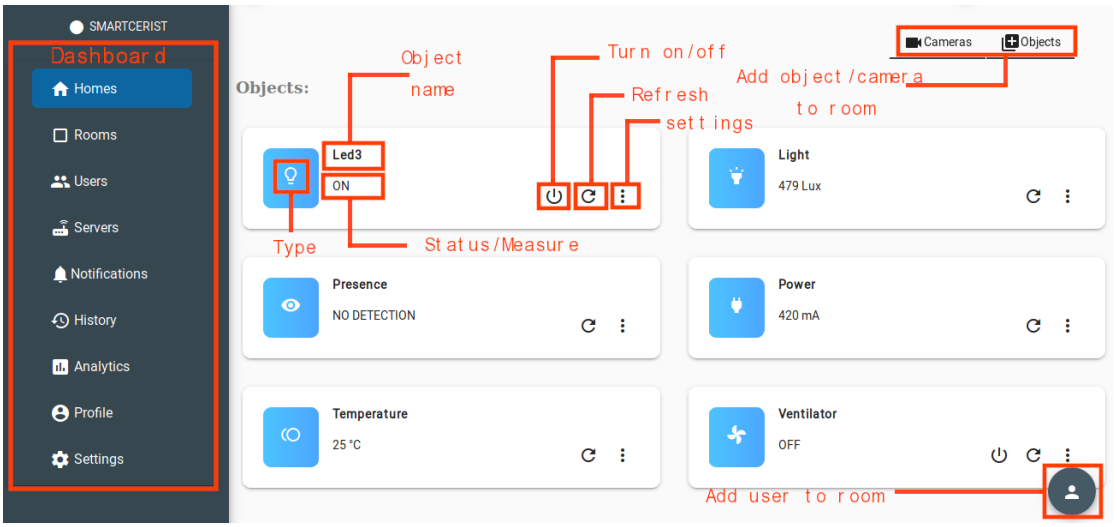


FIGURE 4.9: Web application

4.3.3.3 Mobile Client

The mobile client is implemented with Java, while the communication with objects is achieved using Californium, an implementation of CoAP with java. Our choice of Node-Coap and Californium CoAP implementations is based on a comparison done by [14], who showed that these implementations offer all CoAP functionalities and extensions. Node-Coap is more suitable for applications WEB, and its implementation in Javascript makes it easier to integrate with WEB applications. For the mobile application, the most suitable implementation for Android is Californium that implements all the features of CoAP. Fig. 4.10 illustrates the mobile application.

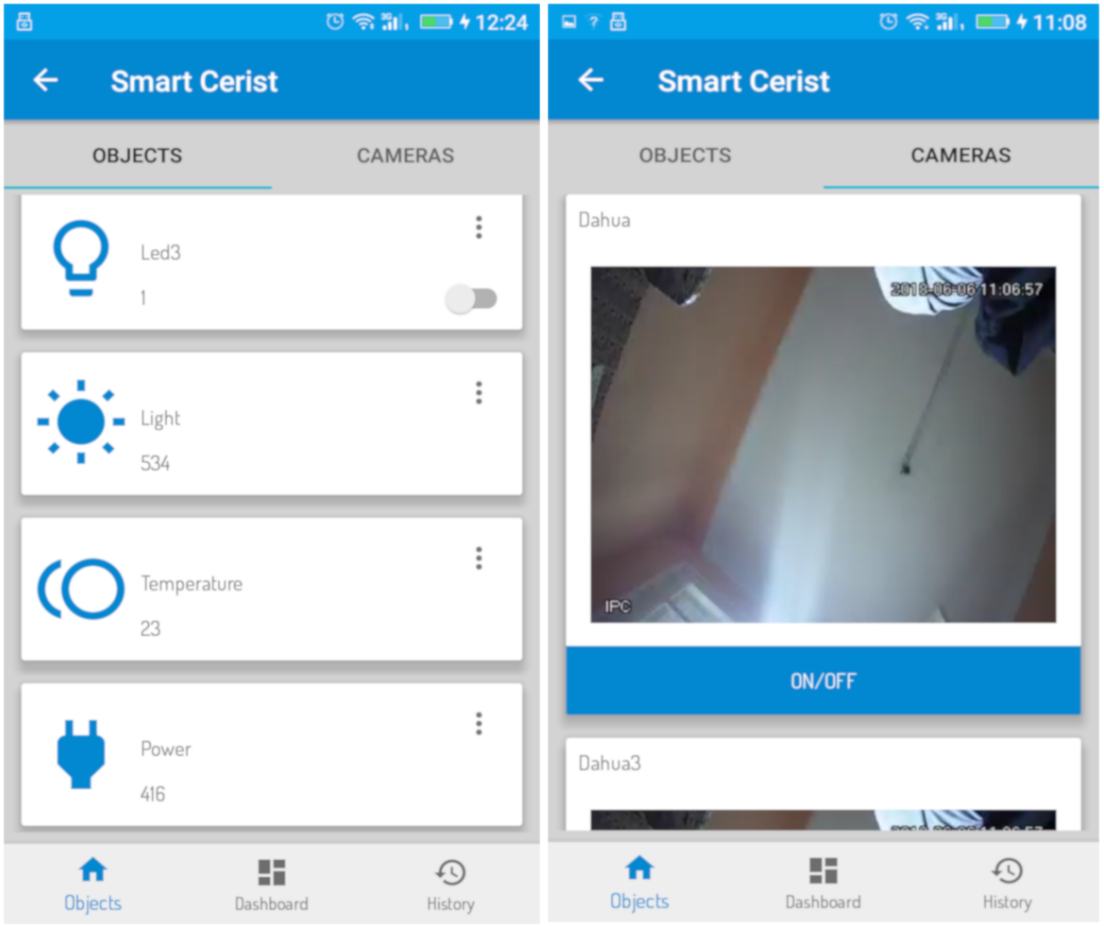


FIGURE 4.10: Mobile application

4.3.3.4 IPv4-IPv6 Translation

Since our system is built on an IoT architecture, the use of IPv6 becomes obvious. However, The problem is that IPv6 is not deployed on a large scale and most devices can only access the IPv4 network. To provide the user with an easy-to-use application that requires no additional configuration, we have opted to translate IPv4 requests into IPv6 requests using a translation mechanism at the application layer. To achieve this mechanism it was necessary that:

- The Raspberry is in dual-stack mode to accept both IPv4 and IPv6 networks.
- The Raspberry is accessible via a public IPv4 address.
- The client (mobile / web) connects with the Raspberry via the IPv4 protocol.

- The Raspberry translates IPv4 requests to IPv6 requests and forwards them to objects.
- The objects respond to the Raspberry with IPv6 packets.
- The Raspberry translates the IPv6 packets coming from the objects to IPv4 packets and transmits them to the client.

When sending the request from the client to the Raspberry server, we add the object address to the body of the request. It will be used as the recipient address by the Raspberry in the IPv6 packet. If the client (web/mobile) has access to IPv6 networks, the user will not need the translation mechanism but queries the objects directly. In this case, the Raspberry acts as a gateway.

4.3.4 Performance Evaluation

Our solution is tested in a smart home environment, which provides a set of sensors and actuators including light, motion, temperature, power, fan and a lamp. The fusion of these heterogeneity data is considered to make the right decision and action in a very short time. Sensors and actuators are connected directly to a Nordic beacons nRF51822, these latter are connected using Bluetooth low energy to a gateway which is a Raspberry Pi. The IPv6 camera is connected to the Internet. Fig. 4.11 below illustrates the hardware used in the tests.

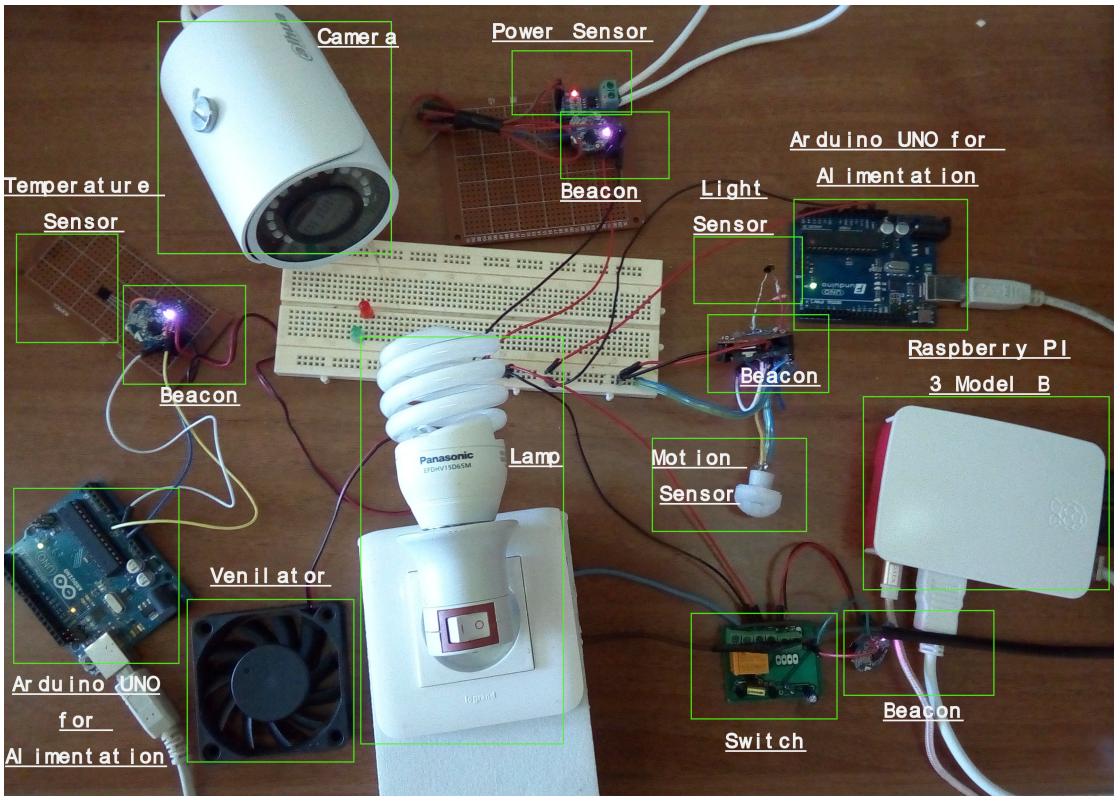


FIGURE 4.11: IoT hardware framework

We evaluate our framework with several tests, we present the response time of objects to query GET in both the web and the mobile applications. Results are shown in Table. 4.1 and Table. 4.2.

TABLE 4.1: Web application requests delay.

Object	Test 1	Test 2	Test 3
Lamp	133 ms	141 ms	119 ms
Motion	141 ms	121 ms	111 ms
power	89 ms	98 ms	138 ms
Light	158 ms	189 ms	138 ms
Temperature	88 ms	109 ms	156 ms

Response time values shown in both mobile and web application do not exceed 190ms. According to [118], to evaluate the performance of our system, the results are perceived as instantaneous and confirm that our system enables real time services.

TABLE 4.2: Mobile application requests delay.

Object	Test 1	Test 2	Test 3
Lamp	120 ms	118 ms	147 ms
Motion	129 ms	96 ms	111 ms
power	112 ms	162 ms	97 ms
Light	78 ms	96 ms	110 ms
Temperature	128 ms	131 ms	138 ms
Ventilator	103 ms	127 ms	127 ms

We varied the number of requests for 4 sensors (Power,light,Temperature,Motion) as depicted in Fig. 4.12. The mean latency are close to each other, and do not exceed 600ms for 500 requests.

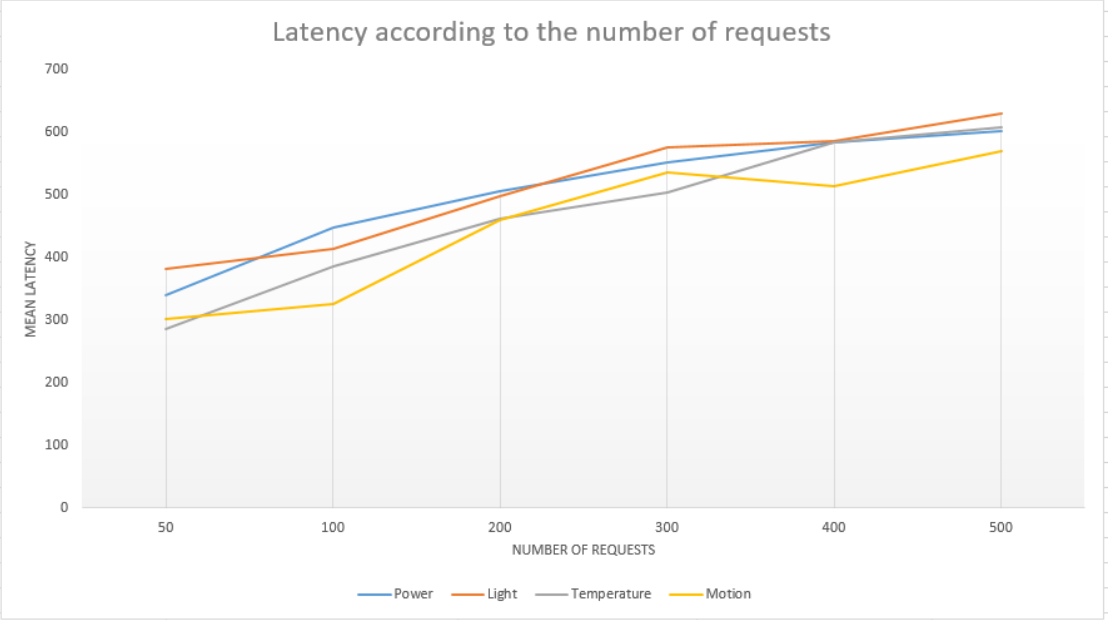


FIGURE 4.12: Mean latency

As shown in Fig. 4.13, the average latency increases slightly with the number of requests. This confirm the effectiveness of our framework in term of latency.

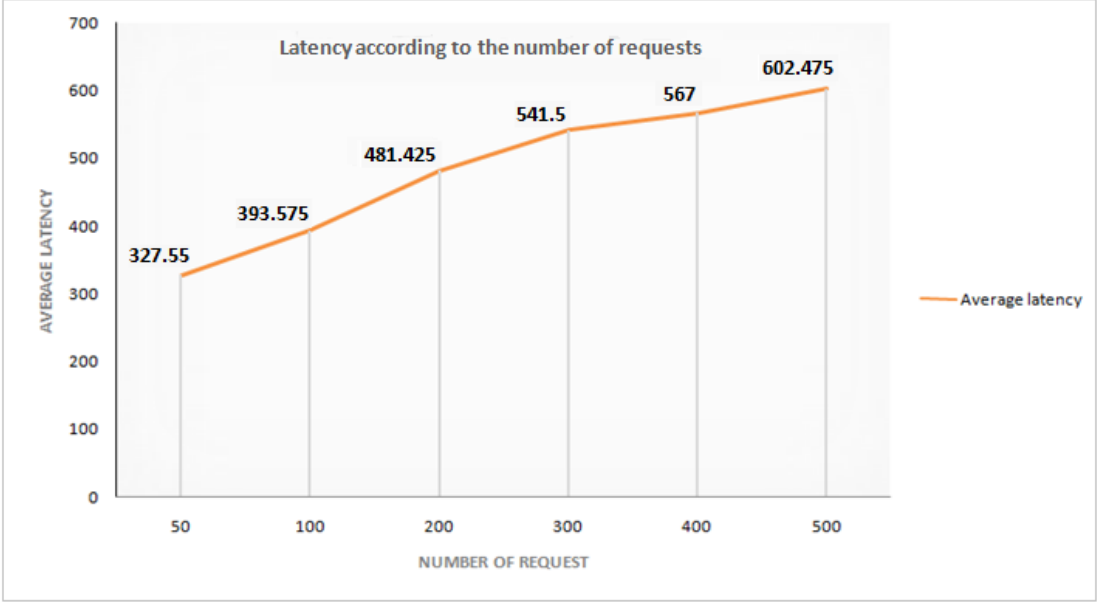


FIGURE 4.13: Average latency

4.4 Conclusion

In this chapter, we proposed two IoT data processing frameworks. The first is application dependent and is baseed on fog computing. The second is hybrid (based on edge, fog, cloud computing) and manages and control data combined from several sensors. It supports IPv6/IPv4 and takes into consideration heterogeneous resources, connectivity/reliability, security and mobility. The proposed frameworks are applied to smart car parks, and smart home environment, respectively. The systems allow to manage and interrogate smart objects (sensors and actuators), the collection of the same or different type of data to follow the evolution of each object as a graph, and to apply rules allowing objects to cooperate and accomplish a specific tasks like turning light on when presence detected or start the ventilation when the temperature exceeds a certain degree. The different rules can be customized by the user. The implemented system allows the user to receive notifications, and to be alerted for any changes in the environment. An access rights management is applied to limit the actions that can be performed by a user. The security of the system in the second solution is ensured by the exploitation of some techniques such as token access, DTLS, encryption of passwords and secure sessions. The framework has been used to develop a smart home control mobile application, which has been extensively tested. The results show

low latency, at the order of few ten of milliseconds for building control over the implemented mobile application, which confirm real time feature of the proposed solution. The frameworks proposed in this chapter are a set of structures that provide the required processing, but a set of principles, practices and tools are still needed. This is by using a methodology to guide processes to achieve a particular goal. In Chapter 5 and Chapter 6 , we propose novel methods of data fusion based on mathematical methods.

Chapter 5

Distributed Particle Filter for Target Tracking and Data Processing in Wireless Sensor Networks

5.1 Introduction

Data Fusion can be seen as the process of combining data or Information to estimate or predict entity states. In the aim of assuring efficiency of fusion at feature level, a novel Distributed Particle Filter for Target Tracking algorithm (DPFTT) is proposed, a new algorithm with new metric. The proposed method address the measurement uncertainty problem and make the particle filter robust to environmental change. Such method can be used in state estimation to fuse data and applied in smart environments and Internet of things applications. This by estimate the kinematic parameters of the target. the aims of our proposal is to calculate the distance between probability densities which is described using Gaussian distribution and generate the optimal importance proposal distribution. The various estimation techniques are compared by computing the estimation root mean square error. The simulation results show that the proposed algorithm, ensures scalability and outperforms the standard particle filter, the improved particle filter based on KLD, and consensus based particle filter algorithm in high noise environment.

The remainder of the chapter is organized as follows. Section 5.2 sketches the solution description. Section 5.3 Formulate the problem and present the system model. Section 5.4 describes the proposed solution, and Section 5.5 describes the simulation results. Finally, Section 5.6 draws conclusions and summarizes the perspectives.

5.2 Solution Description

In order to estimate target tracking in wireless sensor networks, a novel distributed particle filter algorithm for Target Tracking and data processing in wireless sensor networks (DPFTT) is proposed.

The algorithm consist of three phases:

1. Prediction and prior measurement.
2. Importance sampling.
3. Resampling phase.

The particles are selected based on their closeness to the real measurement, in the sense of an appropriate distance. DPFTT calculates similarity between two probability distributions, which based on Jousselme distance [61]. It optimize the number of particles in the resampling phase.

5.3 Problem Formulation

In this section, we review distributed particle filter algorithm and jousseleme similarity distance. Besides that, we propose the new method to calculate the similarity between the true posterior and the proposal distribution. Such measure optimize the number of particles and enhance the particle filter algorithm for object tracking.

5.3.1 State Estimation Technique

The key idea of particle filter is to represent the required posterior density function by a set of random samples with associated weights, and to compute the

estimates based on these samples and weights. The problem major encountered is the large number of samples, this Monte Carlo characterization becomes an equivalent representation of the posterior probability function, and the solution approaches the optimal Bayesian estimate. The particle filter algorithm makes use of an important density, which is a proposed density to represent the posterior one that cannot be exactly computed. Then, samples are drawn from the important density instead of the actual density. A common problem with the particle filter is the degeneracy phenomenon, where after a few states all but one particle will have negligible weight [71]. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation of the posterior density function is almost zero. This problem can be overcome by increasing the number of particles, or more efficiently by approximately selecting the important density. In addition, the use of resampling technique [84] is recommended to avoid the degeneracy of the particles as algorithm. The particle filter algorithm is presented in Algorithm. 5.1.

Algorithm 5.1: particle filter algorithm

1 Prediction :$\widehat{x_n} = f_n(x_{n-1})$
2 prior Measurement: $\widehat{z}_n = h(\widehat{x}_n)$
3 //Importance sampling
4 Draw $\left\{x_n^i\right\}_{i=1}^{Ns} \sim \mathcal{N}(\mu, \sigma)$
5 //Measurement
6 For particle i=1: Ns do
7 Compute likelihood function: $\left\{x_n^i\right\}_{i=1}^{Ns} \propto p(x_n^i|x_{n-1}^i)$ state transition model
8 Weight: $\left\{w_n^i\right\}_{i=1}^{Ns}$ $w_n^i \propto p(z_n^i|x_n^i)$ observation model
9 End For
10 Normalizing: $w_n^i = \frac{w_n^i}{\sum_{i=1}^{Ns} w_n^i}$
11 Resampling: $\left\{x_n^i, , w_n^i\right\}_{i=1}^{Ns}$

5.3.2 System Model

The model based methods for tracking applications generally require two models, namely, state model and measurement model. The former describes the evolution of the state with time while the latter defines the relationship between the noisy observations and state. A state system model for the wireless sensor networks is considered as follows [71]:

1. The state transition model:

$$x_n = Fx_{n-1} + Gu_n$$

where x_n is a vector that denotes unknown states of the dynamic system at time n, F denotes the function that describes the time-evolution of the vector x_n and G is a vector of the state noise.

the state equation is defined in Eq. 5.1 as :

$$F = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, G = \begin{pmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{pmatrix}. \tag{5.1}$$

T_s is the time sampling period , $u_n \sim \mathcal{N}(0, \Sigma_u)$ is a two-dimensional Gaussian noise vector.

2. The measurement model:

$$Z_{k,n} = h_k(x_n) + e_{k,n}$$

where $Z_{k,n}$ is a vector that describes the measurement obtained from sensor k at time instant n, and h_k the measurement function that maps the kinematic vector parameter x_n to the measurement vector $Z_{k,n}$.

The measurement equation of the dynamic system is defined in Eq. 5.2 as :

$$Z_{k,n} = \frac{C}{(x_n - s_{x,k})^2 + (y_n - s_{y,k})^2} + e_{k,n}, \tag{5.2}$$

where $Z_{k,n}$ denotes the measured signal at time n by sensor k, $s_{x,k}$ and $s_{y,k}$ denote the x- and y-axis location of sensor k, C is a constant chosen to calibrate the SNR of the signal received bu the furthest sensor, and $e_{k,n} \sim \mathcal{N}(0, \sigma_e^2)$ is a measurement of Gaussian noise of sensor.

5.3.3 Similarity Distance

The similarity distance is first proposed by Jousselme et al [99] and then applied to weighted averaging combination method by Yong et al. [100]. It can measure

the dissimilarity between two basic belief assignment (bba) defined on the set of all subsets of $\Omega = \{w_1, ..., w_n\}$ called the frame of discernment. The distance between two bodies of evidence, $d_{BOE}(m_1, m_2)$ is defined in Eq. 5.3, for m_1, m_2 on 2^{Ω}:

$$d_{BOE}(m_1, m_2) = \sqrt{\frac{1}{2}(\|m_1\|^2 + \|m_2\|^2 - 2\langle m_1, m_2\rangle)}, \tag{5.3}$$

where $\langle m_1, m_2\rangle$ is the scalar product defined in Eq. 5.4,

$$\langle m_1, m_2\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} m_1(A_i)m_2(A_j)\frac{|A_i \cap A_j|}{|A_i \cup A_j|}, \tag{5.4}$$

where n=$|2^{\Omega}|$.

In order to quantify how much two or more objects are different, we use a distance to measure the similarity between the belief functions.

The distance can measure the conflict degree among evidences effectually. The greater the distance of two bodies of evidence is, the less evidence support each other.

5.4 Solution

We proposed a new method to guarantee a better estimate of the target state using an optimal number of samples. DPFTT is based on similarity distance.

In the first part of this chapter, we have presented different specification of discrete belief function. Unfortunately, these functions do not allow us to manipulate continuous data that can be provided by sensors in different areas like search and rescue problems,classification issues, and data fusion. For that, we are based on the distance of jousselme by calculating the similarity between the probability density functions using normal Gaussian distribution. Belief function can be defined by a finite number of parameters using Smets formalism [119].

5.4.1 Belief Function Associated to a Probability Density

A probability density function (pdf) is considered as an expert belief, it can be defined according to basic belief density which is described using a normal distribution [120]. We defined the distance between two densities in Eq. 5.5,

$$d(f_1, f_2) = \sqrt{\frac{1}{2}(\|f_1\|^2 + \|f_2\|^2 - 2\langle f_1, f_2\rangle)}, \tag{5.5}$$

where $\langle f_1, f_2\rangle$ is defined in Eq. 5.6 as:

$$\langle f_1, f_2\rangle = \int_{-\infty}^{+\infty}\int_{y_i=x_i}^{+\infty}\int_{-\infty}^{+\infty}\int_{y_j=x_j}^{y_j=+\infty} f_1(x_i, y_i) f_2(x_j, y_j)\delta(x_i, y_i, x_j, y_j) dy_j dx_j dy_i dx_i, \tag{5.6}$$

and $\delta(x_i, y_i, x_j, y_j)$ is Lebesgue measure which is an extension of Jaccard measure applied for the intervals in the case of continuous belief functions [119].

This distance can be used between two or more belief functions as Eq. 5.7 :

$$d(f_i, \sigma f) = \frac{1}{n-1}\sum_{j=1, i\neq j}^{n} d(f_i, f_j), \tag{5.7}$$

where σf is defined as a set of belief densities.

5.4.2 Improved Particle Filter based on Similarity Distance

The distributed particle filter algorithm for target tracking is proposed. DPFTT operates in Three steps.
1) In the first one, the samples are initially generated using sensor readings and propagated according to the transition state model $\widehat{x_n} = f_n(x_{n-1})$. When considering the processing noise $\widehat{x_n} = (x_n + q_n)$ where q_n is the additive noise and follows normal distribution. the predicted measurement for sensors derived from $\widehat{x_n}$ is $\widehat{z}_n = h(\widehat{x}_n) = h(x_n + q_n)$ which used as prior information for measurement likelihood.

2) In the second step, a new metric is added in importance sampling phase. We used $f = p_1(z_n/x_n)$ as the objective distribution or proposal and employ $g =$

$p_2(z_n/x_n)$ as the tuning distributions, then the distance is applied using Eq. 5.3. The most likely to choose is the sample whose pdf is nearest to the high likelihood region. We apply this choice using Eq. 5.8 and hence eliminate samples with low importance weights.

$$A = argmin(d(f, g)). \tag{5.8}$$

3) In the last step, the new weight is employed in the resampling phase, and to avoid the degeneracy phenomenon of particle filter algorithm, a suitable measure of the effective sample size N_eff which is introduced in [121], and is defined in Eq. 5.9 as :

$$Neff = \frac{1}{\sum_{i=1}^{N}(w_n^{(i)})^2}, \tag{5.9}$$

where $w_n^{(i)}$ are n normalized weight. DPFTT algorithm is described in Algorithm. 5.2 as follows:

Algorithm 5.2: improved distributed particle filter algorithm

Prediction :$\widehat{x_n} = f_n(x_{n-1})$
prior Measurement: $\widehat{z}_n = h(\widehat{x}_n)$
//Importance sampling
Draw $\{x_n^i\}_{i=1}^{Ns} \sim \mathcal{N}(\mu, Q)$
//Measurement
For particle i=1: Ns do
Compute the similarity distance using Eq. 5.5
Choose the appropriate using Eq. 5.8
Mean: $\widehat{\mu}_{n,k} = \sum_{i=1}^{Ns} w_n^i x_n^i$
Covariance: $\widehat{Q}_{n,k} = \sum_{i=1}^{Ns} w_n^i (x_n^i - \widehat{\mu}_n)(x_n^i - \widehat{\mu}_n)^\intercal$
Compute likelihood function : $\{x_n^i\}_{i=1}^{Ns}$ using Eq. 5.1
Update the weight : $\{w_n^i\}_{i=1}^{Ns} \sim \mathcal{N}_{n,k}(x_n^i, \widehat{\mu}_{n,k}, \widehat{Q}_{n,k})$
End For
Normalizing important weights : $w_n^i = \frac{w_n^i}{\sum_{i=1}^{Ns} w_n^i}$
Resampling with the new samples :
if $Neff = \frac{1}{\sum_{i=1}^{Ns}(w_n^i)^2} < \frac{Ns}{2}$
$\{x_n^i, w_n^i\}_{i=1}^{Ns}$
State Estimation $\widehat{x_n} = \sum_{i=1}^{Ns} w_n^i x_t^i$

5.5 Simulation and Comparative Analysis

For purposes of illustration and visualization of results of the proposed method, we used a simulation scenario to verify the effectiveness of our proposed algorithm. The different patterns are evaluated in terms Root Mean Square Error(RMSE), which indicates the root value of the unbiased covariance. We have computed the RMSE of the estimated target position $(\widehat{x}_n, \widehat{y}_n)$ at each time instant n from all sensors, This using Eq. 5.10.

$$RMSE(n) = \sqrt{\frac{1}{R}\sum_{r=1}^{R}(x_n^{(r)} - \widehat{x}_n^{(r)})^2 + (y_n^{(r)} - \widehat{y}_n^{(r)})^2}, \tag{5.10}$$

where R is the number of simulation runs on which the root mean square is calculated.

In the simulation environment, which is developed in Matlab, A target tracking problem on a two-dimensional plane $(x - y)$ was considered. A sensor network consisting of 100 sensors laid out on a 200 m 200 m field, which is used to collect and process data. The motion of the target is assumed to be constant velocity and the evolution of its kinematic parameters are modeled by: $x_n = Fx_{n-1} + Gu_n$, where the state vector denoted by $x_n = [x_n y_n \dot{x}_n \dot{y}_n]^\intercal$ consists of the xy positions (x_n, y_n) and velocities $(\dot{x}_n, \dot{y}_n)$ of the target. The sampling period Ts is assumed to be equal to 1, and $u_n \sim \mathcal{N}(0, Q)$ is a two-dimensional Gaussian noise vector with covariance of $Q = diag(0.005, 0.005)$. The measurements obtained by the sensors are expressed as a function of the distance of the sensors from the location of the target is given by the measurement equation described in system model. where C=570 and $e_{k,n} \sim \mathcal{N}(0, \sigma_e^2)$ is a measurement of Gaussian noise of sensor, with unity variance $\sigma_e^2 = 1$.

To evaluate the performance of DPFTT and to compare it to the other algorithms such as the standard particle filter algorithm DPF, improved particle filter using KLD (IKLD) in [79], and improved particle filter with exchange data (EXCH) in [71], we used the same runtime environment and the same simulation parameters, we executed our proposed algorithm, comparing it with the algorithms mentioned. The calculation results of RMSE of the estimated target position as time instant evolves from 1 to 60 are depicted in Fig. 5.1.

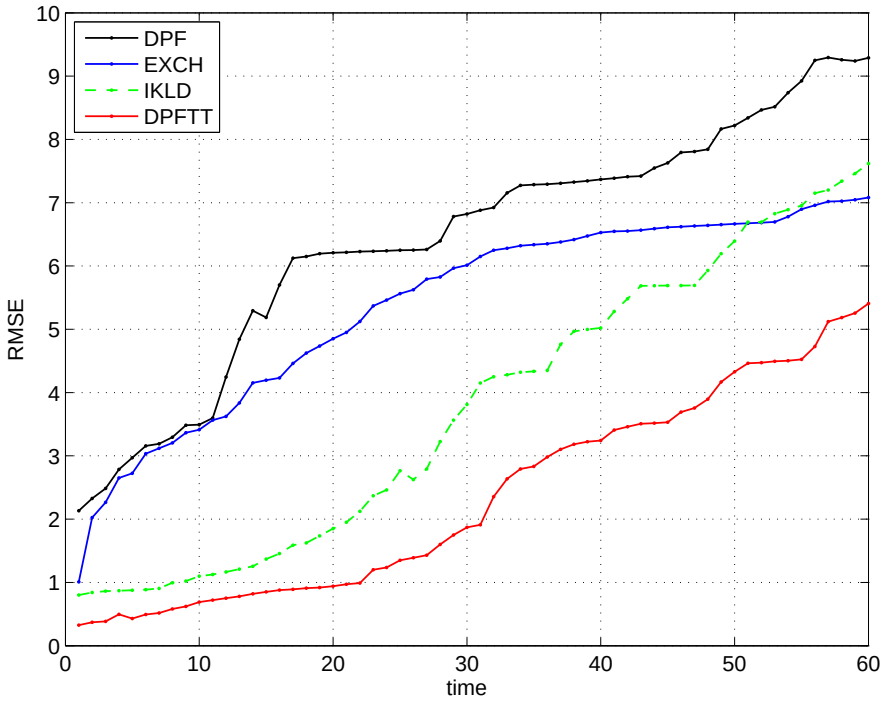


FIGURE 5.1: RMSE

According to Fig. 5.1, the RMSE, for DPFTT is less than the original particle filter DPF. Additionally, the DPFTT have about 1m less than IKLD and more than 1m less than EXCH in the whole of time. The max error, which is the worst case for estimation, in the DPF is more than 9m and even between 7m and 8m in EXCH and IKLD. However, DPFTT have about less than 5,4 m,which is more precise than the other solutions, and hence indicates that our adaptation method can improve the measurement likelihood and estimation accuracy. We also compare estimation performance of DPFTT with varying the sensors number. In order to test the scalability, we executed our solution by varying the number of sensors, we carried out tests based on RMSE. The results of this proposal are represented in Fig. 5.2

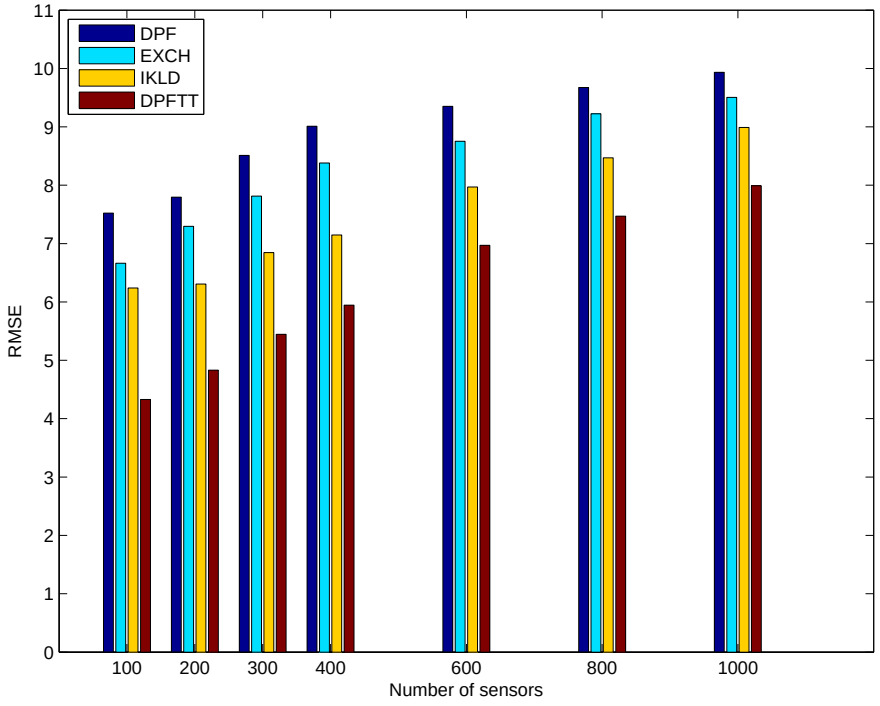


FIGURE 5.2: RMSE with variations in number of sensors

As shown in Fig. 5.2, The RMSE of DPFTT is the lowest compared to DPF,EXCH and IKLD, with the increased sensors number, we can still see that our solution remains reliable and provides the best estimate among the other solutions because despite the change of environment (number of sensors) it ensures good results, scaling and so it is reliable.

5.6 Conclusion

In this chapter, we proposed an improved distributed particle filter algorithm is proposed to deal with target tracking in wireless sensors networks. It increase the estimation accuracy of the particle filter, enhance the efficiency of the particle sampling and improve the estimation performance. The proposed solution is based on distributed particle filter and the Sequential Monte Carlo algorithm for weight adjustment. Adding to it a novel similarity calculation method between two probability density functions using jousselme distance, such measure optimize the number of particles used in resampling phase. A simulation study that compares the proposed solution with two state-of-the-art solutions shows the superiority of the proposed approach in Root Mean Square Error and ensures scalability. In

perspective, we plan to propose an alternative for reading real data by simulating targets tracking, to test the solution in a real deployment scenario, and to study the efficiency of the solution with high volume of heterogeneous data. DPFTT is proposed in the feature level of data fusion, in the next chapter, we proposed efficient method of data fusion in decisional level.

Chapter 6

DFIOT: Data Fusion for Internet of Things

6.1 Introduction

Our solution presented in the previous chapter is state estimation technique, which is defined in feature level of data fusion. We consider in this chapter the decision fusion, where the fusion supplies decisional information. Decision fusion methods provide a formalism for combining evidence according to the probability theory rules, where uncertainty is represented using the conditional probability terms that describe beliefs.

The D-S theory [87] is largely used for uncertainty reasoning, which allows processing uncertain or imprecise information without prior knowledge. This can be well used in IoT environment. It supports the representation of both imprecision and uncertainty, and it allows deriving the probabilities of a collection of hypothesis while dealing with missed information. This can be helpful in such heterogeneous IoT data. However, under situations where the evidence highly conflicts, it may obtain counterintuitive results. This problem is tackled in this chapter in which we first present the literature on decision fusion methods, and then propose a taxonomy of D-S approaches. The most common methods are reviewed by exploring benefits and challenges. A new efficient method is then developed, which emphasizes the importance of reducing the uncertainty of the measurements and conflicts in data fusion. It takes into consideration the contextual IoT parameters while reducing the degree of uncertainty. It also enhance the entropy of Deng,

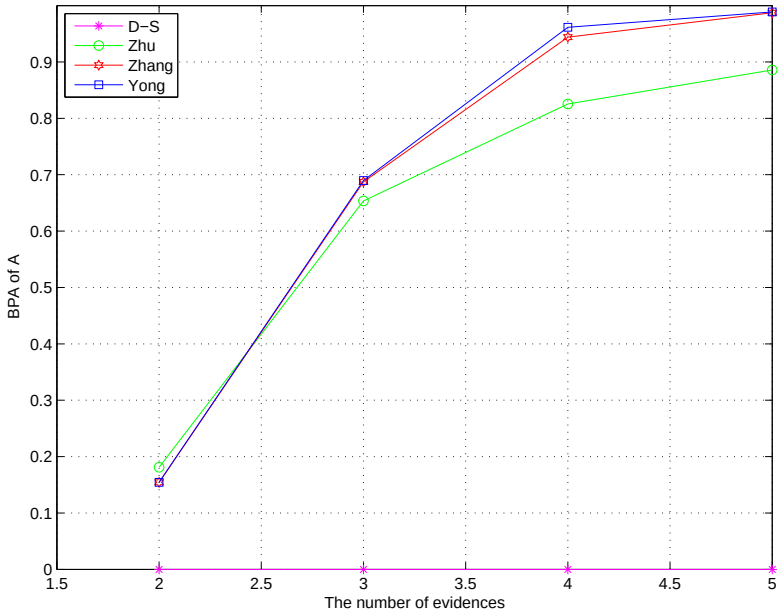
by selecting a sensor report which has a big information volume and well supported by the others sensors. This by exploring similarity between evidence, and hence enhances credibility. Results show that the proposed solution outperforms all the above mentioned methods in terms of reliability, accuracy, and conflict management.

6.2 Solution Description

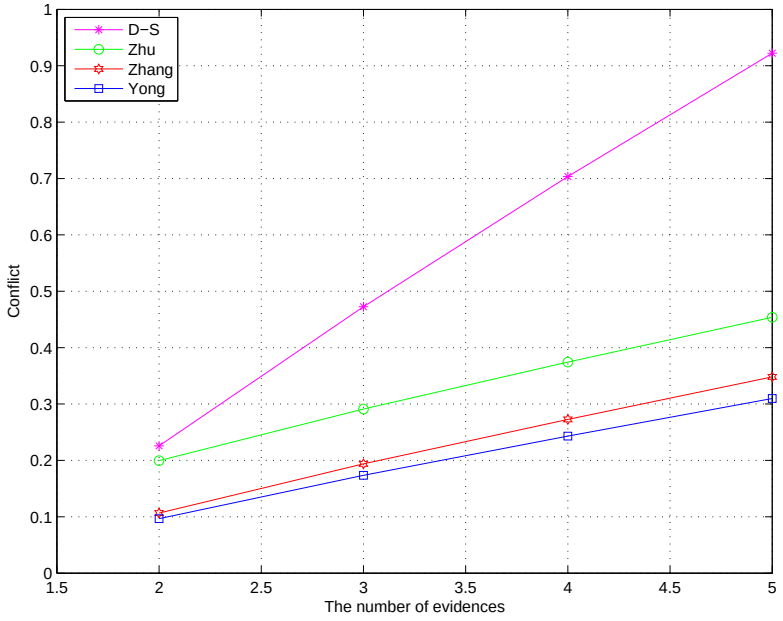
In this chapter, a weighted evidence combination method is used. It is based on weight which represents the degree of confidence that is given to a data source. The method is largely used for uncertainty measure [109] to handle conflicting evidence combination and to take into consideration heterogeneous data in IoT [101, 102]. Our solution addresses the features of IoT data fusion which are: i) the uncertainty, as data provided by sensors, is always subjected to some level of uncertainty and inconsistency. Data fusion algorithms reduce uncertainty by combining data from several sources. ii) Conflicts to present quality data to users, which is critical to resolve conflicts and discover parameters' values that reflect the real world. iii) Energy consumption has always been challenging in wireless sensors networks, and consequently in IoT. To justify the choice of the weighted method upon which we rely for developing the proposed solution, a comparison between several state-of-the-art methods in each subcategory has been carried out. The results are presented in the following.

6.2.1 Comparison Between Weighted Methods

We implemented and compared the methods based on reliability, which are all based on jousselme distance. A benchmark numerical example has been used [101]. The results are given in Fig. 6.1(a) and Fig. 6.1(b). They show that both Yong and Zhang methods give approximately the same BPA of hypothesis A (up to 98%). Yong method, however, has less conflict and increases its performance as the number of evidence goes up.



(a) BPA

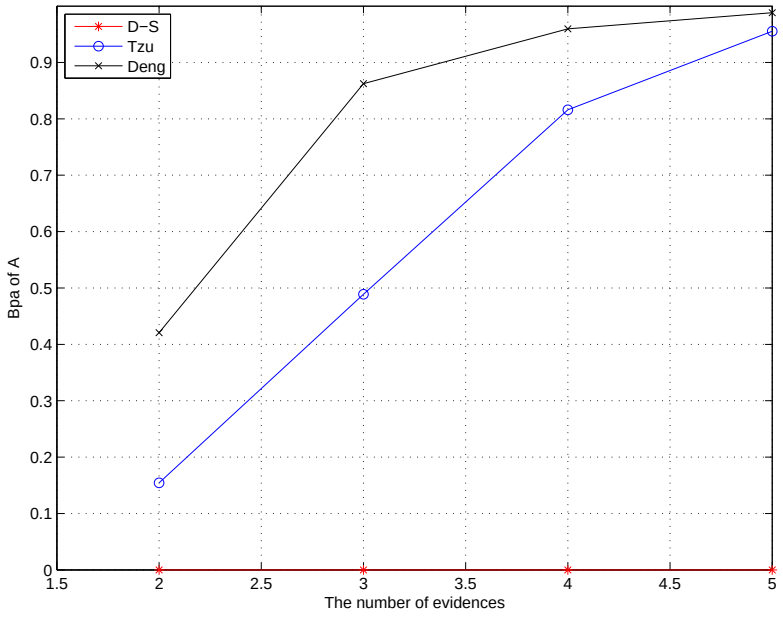


(b) Conflict

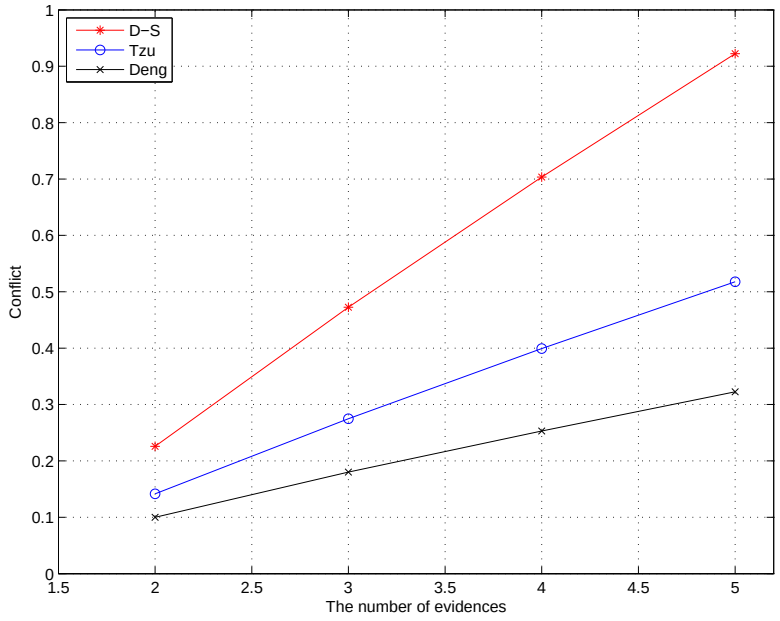
FIGURE 6.1: Comparison between weighted data fusion methods based on reliability

We implement also the second category of data fusion methods based on the amount of information, and compare them using a benchmark numerical example [106]. The results are depicted in Fig. 6.2(a) and Fig. 6.2(b). Deng entropy has clearly the best results in terms of accuracy (BPA) and reaches up to 98%, while D-S's BPA is equal to 0. Deng also has the best performance in terms, e.g.,

it remains at 0.32 for 5 bodies of evidence while the other solutions exceed 0.50. Based on these results, we use Yong method to calculate the credibility of evidence and Deng entropy to capture the information volume when combining evidence.



(a) BPA



(b) Conflict

FIGURE 6.2: Comparison between weighted data fusion methods based on amount of information

The aim is to select the best improved method and use it to calculate the credibility degree of evidence in the first category and the uncertainty degree in the second

one. The simulation results justify our choice of Yong and Deng entropy.

6.2.2 DFIOT Steps

DFIOT is based on D-S theory and Deng entropy. It proposes three methods IDeng, WDST and WDSD, to improve the information processing. IDeng is an improved Deng method that gives more importance to a sensor which has low uncertainty, i.e., more volume of information, and less evidence distance vs. other sensors. That is, if a sensor report has a big information volume and a less evidence distance, it will be well supported by other sensors and thus will have a higher weight proportion. Otherwise, a small weight proportion will be assigned to sensors with conflicting readings (having high evidence distance) or with a low volume of information. The contextual parameters used in DFIOT are, 1) the lifetime of sensed data, which is represented by Weighted Dempster Shafer method based on Time (WDST), and 2) the distance between sensors and the entity which is represented by Weighted Dempster Shafer method based on Distance (WDSD). DFIOT includes four steps as illustrated in Fig. 6.3, and detailed in the following:

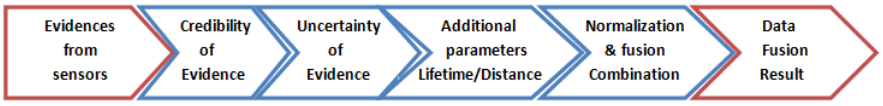


FIGURE 6.3: The flowchart of DFIOT method

6.2.2.1 Calculate the Credibility Degree of Evidence

For every piece of data collected by sensors, Eq. 3.6 is used to calculate the distance between every two bodies of evidence, and Eq. 3.8 to obtain the support degree

of each evidence. Algorithm. 6.1 describes the calculation of evidence credibility.

Algorithm 6.1: Credibility computation of each evidence

Input : <m1,m2,m3,.> [vector of evidences]
N: number of evidence
$T = h1, h2, h3, ...$ frame of discernment.

Output: CRD vector

$CRD = zeros(N) : Initialization$
Sum=0;
SumCRD=0
Begin
z **for** $i = 1$ *to* N **do**
    for $j = 1$ *to* N **do**
        if $i \neq j$ **then**
            $Sum = Sum + 1 - d_{BPA}(mi, mj)$
        end if
    endfor
    CRD[i]=Sum;
    SumCrd=SumCrd+CRD[i];
    Sum=0;
    endfor
for $i = 1; i <= N; i++$ **do**
    $CRD[i] = CRD[i]/SumCrd$
    endfor
Return CRD vector
End

6.2.2.2 Calculation of the Uncertainty Degree of Evidence

We propose an Improved Deng method (IDeng) to calculate the information volume (degree of uncertainty) associated with each evidence. The Deng entropy presented in Chapter 3 is used. In fact, each evidence suffers from a degree of uncertainty that influences on the degree of its involvement in the final combination.

Three cases can be identified: i) Evidence has a high degree of uncertainty (low information volume); its weight must be set to a small value (case of uncertainty). ii) Evidence has a low degree of uncertainty and significant evidence distance

with the other evidence; its weight must be set to a small value (case of negative certainty). iii) Evidence has a low degree of uncertainty and low evidence distance; its weight must be set to a high value (case of positive certainty). The information volume of sensed data $I(SD)$ will be calculated for each evidence in two steps as follows:

First, the entropy of each evidence is calculated with Eq. 3.10. Second, the distance of evidence is verified in terms of decision with other bodies of evidence. To determine if the certainty is positive or negative, the sum of the distance is calculated and the farthest evidence is ignored. If the sum reaches a certain threshold then the certainty is considered negative, otherwise it is considered positive. This is explained in Algorithm. 6.2. The final weight for each evidence is calculated using Eq. 6.1.

$$ED = 1 - \frac{ED}{sum(ED)}. \tag{6.1}$$

where $sum(ED)$ is the sum of all evidence.

Algorithm 6.2: Information volume computation of each evidence

Input : <m1,m2,m3,.> [vector of evidences]
N: evidence number
$T = h1, h2, h3, \cdots$ frame of discernment.
Output: ED vector; information volume of sensors

```
ED= zeros(N); NC= zeros(N); booleen , NC:Negative Certainty
SumDisatance = 0; MaxDistance = 0;
Begin for i = 1 to N do
    SumDistance=<Sum of evidences>
    MaxDistance=<the largest distance between evidence i and the others>
    ED[i]=DengEntropy(mi);
    if SumDistance − MaxDistance) > 0.5(N − 2) then
        NC[i] = true;
    endif
    done
for each evidence i do
    ED = ED ∗ N/Sum of ED
    if NC[i] = true then
        RetunED = ED/2;
    endif
    endfor
Return the vector ED
```

6.2.2.3 Addition of Contextual Parameters

In IoT applications, a large collection of sensors, devices, and users provide a large amount of information in different contexts. This information is usually prone to errors and lacks reliability and credibility. The information to be used must be analyzed, substantiated and motivated. Therefore, merging contexts while increasing the confidence to bring new information, and giving a complete view of the environment is important when fusing data in IoT.

To enhance the quality of data fusion in large heterogeneous and distributed wireless sensors networks used in IoT applications, we consider contextual parameters that are essential for any type of IoT application. These parameters measure the degree of conformity of the IoT environment as perceived by the measuring device. The first parameter is sensor confidence distance, which is a quality parameter that measures the accuracy of the information in a context object. The quality of the decision made by a sensor is strongly affected by the resolution of the space, namely, the distance between the sensor and the entity in question. The second parameter that has been taken into account is the time validity or lifetime of the information. In fact, the information taken at time t is more important than information taken at $t-1$. Furthermore, the information becomes invalid after a certain time, which depends on the context of the application.

Lifetime of Sensed Data: A new parameter is added into WDST that measures the validity of contextual information. The lifetime of sensed data(SD) is normalized by Eq. 6.2

$$Lifetime(SD) = 1 - \frac{Age(SD)}{\Delta T_{max}}. \tag{6.2}$$

where ΔT_{max}, the maximum lifetime or total duration of the information when we recover all sensed data, Age(SD) represents the lifetime of the observation, which is given by Eq. 6.3 :

$$Age(SD) = T - T_{mes}(SD) \tag{6.3}$$

where T is the current time and $T_{mes}(SD)$ the time when SD was collected.

Distance between Sensors and the Entity: In the WDSD, the quality of the decision−making is strongly affected by the resolution of the space, i.e., the

distance between the sensor and the monitored entity. The normalized function is given in Eq. 6.4 is used. The following function:

$$Dis(S) = 1 - \frac{D(S,E)}{D_{max}} * \delta. \tag{6.4}$$

where $D(S,E)$ denotes the distance between the sensor S and the entity E, $Dmax$ the maximum distance upon which the sensors observation can be trusted, and δ the accuracy of a sensor measured on the basis of statistical estimation.

6.2.2.4 Normalization of the Weights and Data Fusion Combination

For each evidence, i, the weight W_i is defined with Eq. 6.5,

$$W_i = CRD(i) * ED(i) * Lifetime(i) * Dis(i). \tag{6.5}$$

Assume there are k evidence; the normalized weight is given by Eq. 6.6.

$$\widetilde{W_i} = \frac{W_i}{\sum_{j=1}^{k} W_j} (i = 1, 2, ..., k). \tag{6.6}$$

We use the weights obtained in Eq. 6.6 to calculate the BPA. The weighted evidence are then combined using orthogonal (Eq. 6.7). Notice that the Dempster's rule of combination has the attractive property of commutativity and associativity [89].

$$M = m_1 \oplus m_2 \oplus ... \oplus m_k. \tag{6.7}$$

6.3 Simulation and Comparative Analysis

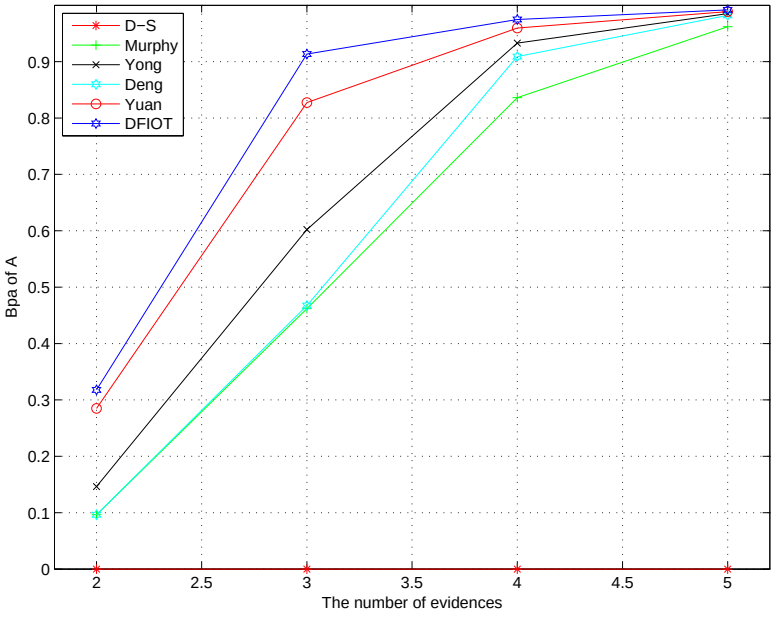
A numerical analysis with MATLAB [122] is presented in this section. The proposed solution is compared with five state-of-the-art methods, including D-S [92], Murphy [98], Yong [100], Deng [106], and Yuan [109]. These methods are the most efficient from the literature for conflict management, and thus the most relevant for comparison. A benchmark numerical example[101] has been used. In a system of automatic target recognition $\{A, B, C\}$ based on different types of sensors

(CCD, sound, infrared, radar and ESM), assume that the current target is A and the number of sensors is five. The system has collected five bodies of evidence, the results obtained are shown in Table. 6.1.

TABLE 6.1: BPAs of the example

	A	B	C	A,C
S1:m1(.)	0.41	0.29	0.3	0
S2:m2(.)	0	0.9	0.1	0
S3:m3(.)	0.58	0.07	0	0.35
S4:m4(.)	0.55	0.1	0	0.35
S5:m5(.)	0.6	0.1	0	0.3

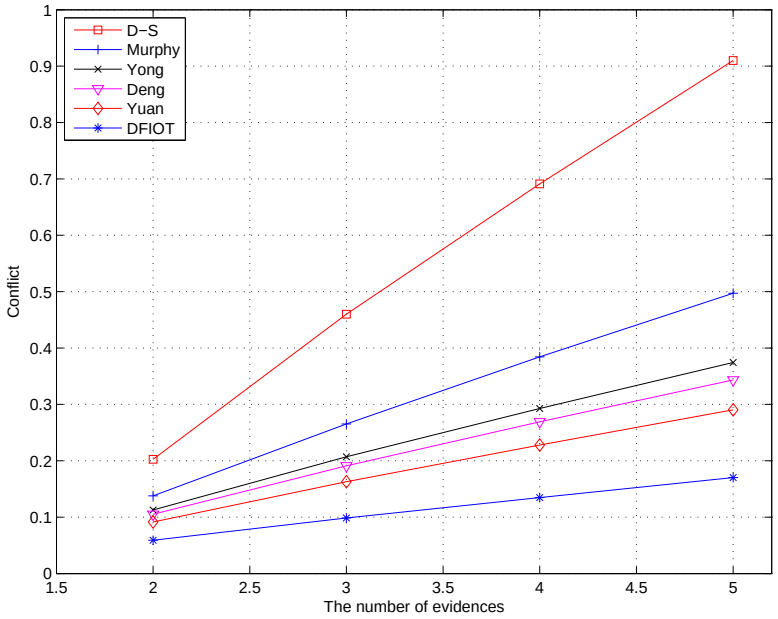
Notice for the BPAs given above that the detection of $S2$ is abnormal. This may lead to a contradictory result after fusion. As the information concerning the lifetime of sensed data, and the distance between sensor and entity are required in the benchmark example, we used our solution without considering contextual parameters. Two metrics are used in the simulation, 1) certainty in the decision (BPA), and 2) the conflict between evidence. Table. 6.2 depicts the fusion results when varying the combination rules and the number of evidence. The calculation process about the last column of the proposed method is given.



(a) The belief value (BPA) of the target A

TABLE 6.2: The results of different combination rules

Method	Fusion Results			
	m1,m2	m1,m2,m3	m1,...,m4	m1,...,m5
D-S	m(A)=0 m(B)=0.8969 m(C)=0.1031	m(A)=0 m(B)=0.6575 m(C)=0.3425	m(A)=0 m(B)=0.3323 m(C)=0.6679	m(A)=0 m(B)=0.1422 m(C)=0.8578
Murphy	m(A)=0.0964 m(B)=0.8119 m(C)=0.0917 m(AC)=0	m(A)=0.4619 m(B)=0.4498 m(C)=0.0792 m(AC)=0.0090	m(A)=0.8362 m(B)=0.1147 m(C)=0.0410 m(AC)=0.0081	m(A)=0.9620 m(B)=0.0210 m(C)=0.0138 m(AC)=0.0032
Yong	m(A)=0.1463 m(B)=0.7620 m(C)=0.0917 m(AC)=0	m(A)=0.6021 m(B)=0.2907 m(C)=0.0990 m(AC)=0.0082	m(A)=0.9330 m(B)=0.0225 m(C)=0.0353 m(AC)=0.0092	m(A)=0.9851 m(B)=0.0017 m(C)=0.0096 m(AC)=0.0035
Deng	m(A)=0.0964 m(B)=0.8119 m(C)=0.0917 m(AC)=0	m(A)=0.4674 m(B)=0.4054 m(C)=0.0888 m(AC)=0.0084	m(A)=0.9089 m(B)=0.0444 m(C)=0.0379 m(AC)=0.0089	m(A)=0.9820 m(B)=0.0008 m(C)=0.0089 m(AC)=0.0036
Yuan	m(A)=0.2849 m(B)=0.5306 m(C)=0.1845 m(AC)=0	m(A)=0.8274 m(B)=0.0609 m(C)=0.0986 m(AC)=0.0131	m(A)=0.9596 m(B)=0.0032 m(C)=0.0267 m(AC)=0.0106	m(A)=0.9886 m(B)=0.0002 m(C)=0.0072 m(AC)=0.0039
DFIOT	m(A)=0.3178 m(B)=0.5233 m(C)=0.1589 m(AC)=0	m(A)=0.9134 m(B)=0.0039 m(C)=0.0395 m(AC)=0.0333	m(A)=0.9748 m(B)=0.0002 m(C)=0.0066 m(AC)=0.0183	m(A)=0.9918 m(B)=0.0001 m(C)=0.0061 m(AC)=0.0020



(b) The value of conflict between evidences

FIGURE 6.4: The fusion results comparison between DFIOT and different rules

Fig. 6.4(a) shows the evolution of belief's value assigned to the target A (the right decision) after each combination of five bodies of evidence for the compared methods. Fig. 6.4(a) shows that the accuracy (BPA) of the proposed solution is always superior to all the other solutions and reaches up to 99.18%. More importantly, while all solutions converge to BPA values beyond 95%, DFIOT grows very fast. It exceeds 90% with only three bodies of evidence, while Yuan's method reaches about 82% and the others remain below 60%. DFIOT also provides the best performance for conflict that was smoothly increasing between 0.058 and 0.170 as shown in Fig. 6.4(b), while all the other solutions exceed 0.29 for five bodies of evidence (some go up to 0.9). These results confirm the efficiency of the proposed method and the effectiveness of considering conflict, information volume.

6.4 Experimental Performance Evaluation

The model is evaluated through an extensive set of experiments realized in (CERIST-ALGERIA) research center lab in the context of IoT and smart building project. In addition to the simulation study presented in the previous section. A real dataset collected from a testbed has been used for further investigation on the performance of DFIOT in comparison with state-of-the-art solutions. An IoT based building automation application has been considered in the office scenario, where presence and ambient light are permanently monitored with IoT enabled wireless sensors to optimize lighting and energy control of appliances and HVAC systems. 4 sensors have been placed on the ceiling of an office, including 3 PIR sensors and a light sensor. The reason behind using three sensors is the need to detect even small movements when the office is occupied (e.g., typesetting on the keyboard, head movements, etc.), which usually occur at the desk space. No placement of a single sensor can guarantee total coverage of this space. Optimal positions of the PIR sensors have been calculated using an integer linear programming (ILP) model and CPLEX solver, where the space has been uniformly split into equal distance grids. The outcome for a $30cm$ granularity unit provides optimal number of sensors that cover the whole office space (centers of cells in the space) to be four, with the positions depicted in Fig. 6.5.

A handy device at the entrance has been added as a ground truth sensor for the presence, with buttons the occupants are asked to push on for every entrance/exit during the experiment. Every node acts as a source and sends data periodically to the central station that stores the data in a dataset. We define 4 hypothesis

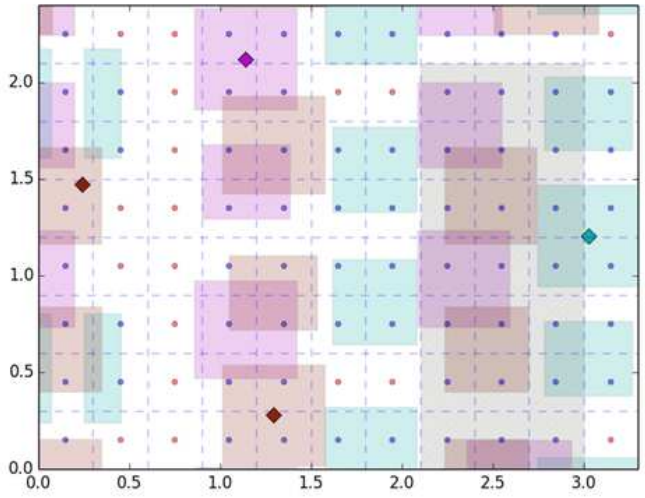


FIGURE 6.5: Deployment of sensors in office

$(H1, H2, H3, H4)$ as follows. $H1$: office occupied and light value is more than $580lux$, $H2$: office empty and light value is more than $580lux$, $H3$: office occupied and light value is not exceeding $580lux$. $H4$: office empty and light value is not exceeding $580lux$. The frame of discernment is $o = H1, H2, H3, H4$. Without loss of generality, we use a simple scenario; light control, where the data fusion method is applied to make a decision of switching on/off the light. The BPA has been calculated using the mean values in each state and the environment has been simulated with a standard error and confidence interval of 98%. In the following, we consider the situations when hypothesis $H1$ and $H2$ are verified. The ground truth has been used to filter out entries in the dataset where each hypothesis is verified. In Table. 6.3, the system has collected bodies of evidence when hypothesis $H1$ is verified. This includes PIR sensors, light sensor, and a ground truth sensor:

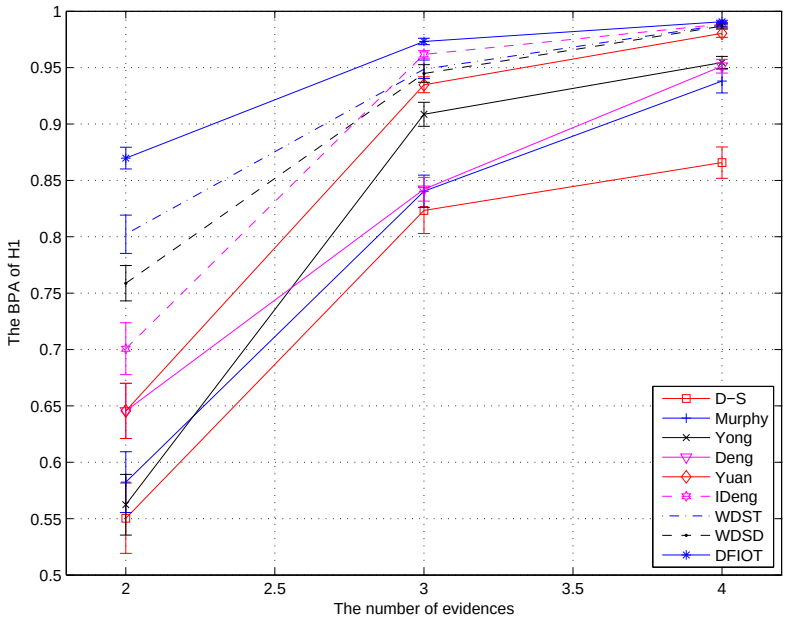
TABLE 6.3: BPAs of the solution

	H1	H2	H3	H4
S1:m1(.)	0.72	0.17	0.10	0
S2:m2(.)	0.69	0.08	0.22	0.01
S3:m3(.)	0.81	0.06	0.11	0.02
S4:m4(.)	0.83	0.07	0.09	0

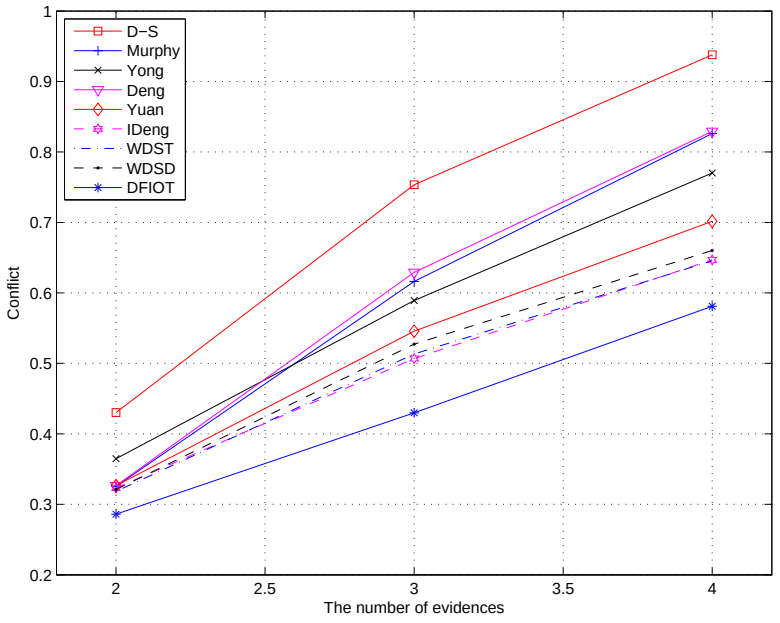
The BPA of Hypothesis and the conflict degree are presented vs. the variation of evidence numbers.

6.4.1 BPA/Conflict of Hypothesis H1

We illustrate the three proposed methods (IDeng, WDST and WDSD) separately to investigate the efficiency of each method of DFIOT.



(a) BPA of hypothesis H1



(b) Conflict of hypothesis H1

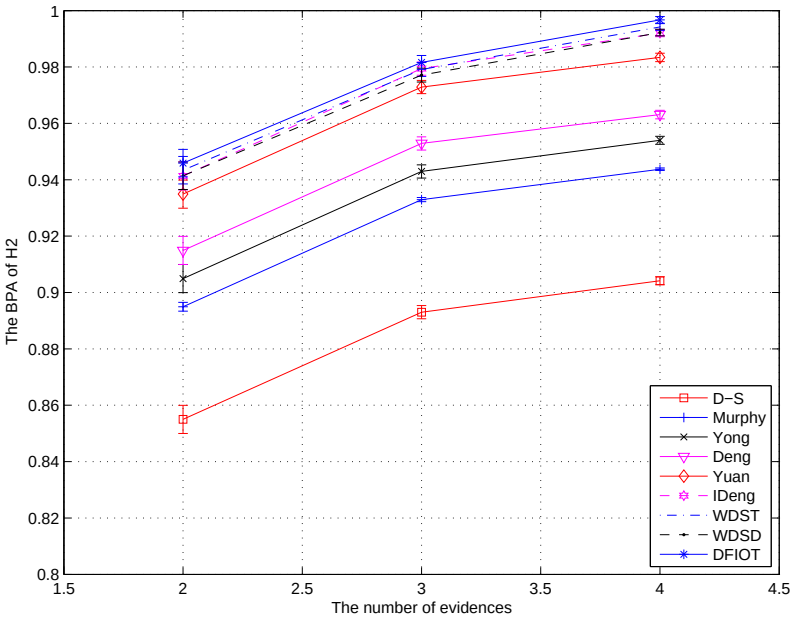
FIGURE 6.6: The fusion results for Hypothesis H1

In D-S evidence theory, the BBA of H1 is 87%, and it decreases gradually with the increase of evidence. The conflict hugely increases and reaches 0.92. In Yuan method, the belief degree of H1 is 98%, while the new method has a higher belief degree of 99.18%. The main reason is that the proposed method takes into consideration the dynamic reliability measured by evidence distance and entropy, it also enhance the Deng entropy and add a new parameters, which decreases the conflict less than 0.6. These results improve accuracy and confirm the efficiency of DFIOT. Each parameter added considerably increases the BPA. The results presented in Fig. 6.6(a) and Fig. 6.6(b) confirm those presented in the previous section and the superiority of the proposed solutions. Similarly, BPA of DFIOT and all its versions grows much faster than the other solutions, and have lower conflict values.

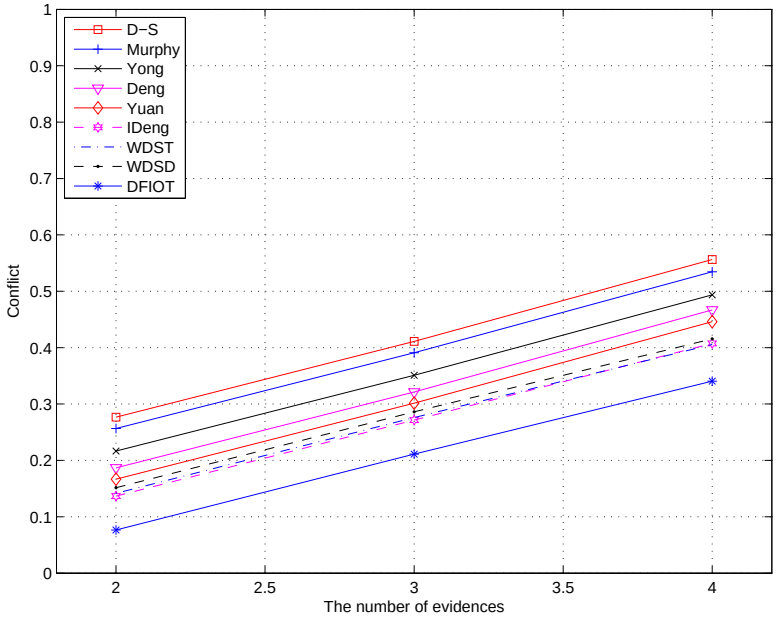
6.4.2 BPA/Conflict of Hypothesis H2

The hypothesis, in this case, is H2 that is when we have no presence detection (empty office) and light exceeding $580lux$. The results are depicted in Fig. 6.7(a) and Fig. 6.7(b) and confirm the superiority of our method. The results are very similar to those obtained for hypothesis H1, except that Yuan provides a bit better performance in terms of BPA (compared to its performance in hypothesis H1), but still clearly lower than DFIOT.

A sensor may be misled by many factors and provide wrong evidence, and such abnormal measurement can generate a conflicting mass during the evidence combination which leads to the conflict. As shown in the simulation section, the classical Dempster's rule cannot support the correct hypothesis H1 and cannot eliminate the conflict between evidence. and a wrong final result. With incremental evidence, our solution provides better results.



(a) BPA of hypothesis H2



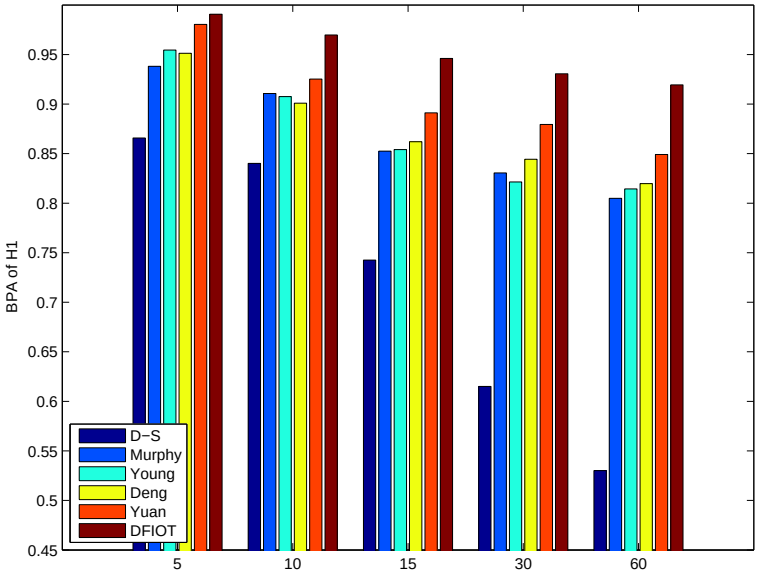
(b) Conflict of Hypothesis H2

FIGURE 6.7: The fusion results comparison with hypothesis H2

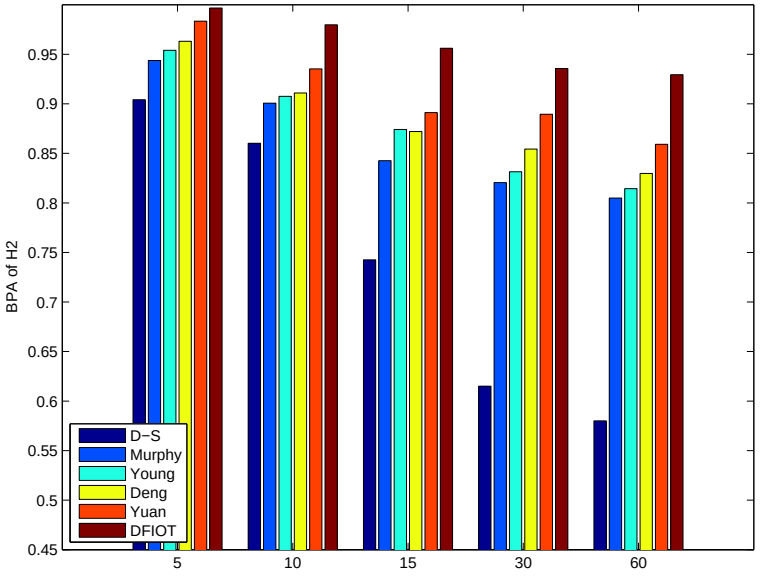
6.4.3 Impact of Data Fusion Period

6.4.3.1 Impact on BPA

We investigated the impact of the data fusion period on the BPA. Results in Fig. 6.8(a) shows that the proposed solution provides the best values and is less affected by increasing this period compared to the other solutions. It remains beyond 90% even for as a high period as $60min$. Similarly in Fig. 6.8(b), when Hypothesis $H2$ has applied the results show the effectiveness of our method when reducing the period of data fusion. While the other solutions are influenced considerably by the increase of the data fusion period, especially in D-S method where the BPA is less than 50% in both cases of H1 and H2, which leads to a wrong decision (switching on/off the light).



(a) Hypothesis H1



(b) Hypothesis H2

FIGURE 6.8: Data fusion period

6.4.3.2 Energy Gain

We investigated the impact of this data fusion on the application performance, in our case energy gain is used as the application related metric. It is defined in Eq. 6.8.

$$Gain(A) = \frac{\sum_{i=1}^{n} \alpha_i}{\sum_{j=1}^{k} \gamma_j}. \tag{6.8}$$

where α_i represents the total period when the office is unoccupied (hypothesis $H2$ and hypothesis $H4$ are verified.), and γ_j represents the total period of data fusion in a day. As shown in Fig. 6.9, the gain increases with the increase of the fusion frequency (decreasing the period). The increase is fast for values bellow $30min$ and reaches up to 90% for a data fusion period of $5min$, which is a reasonable period.

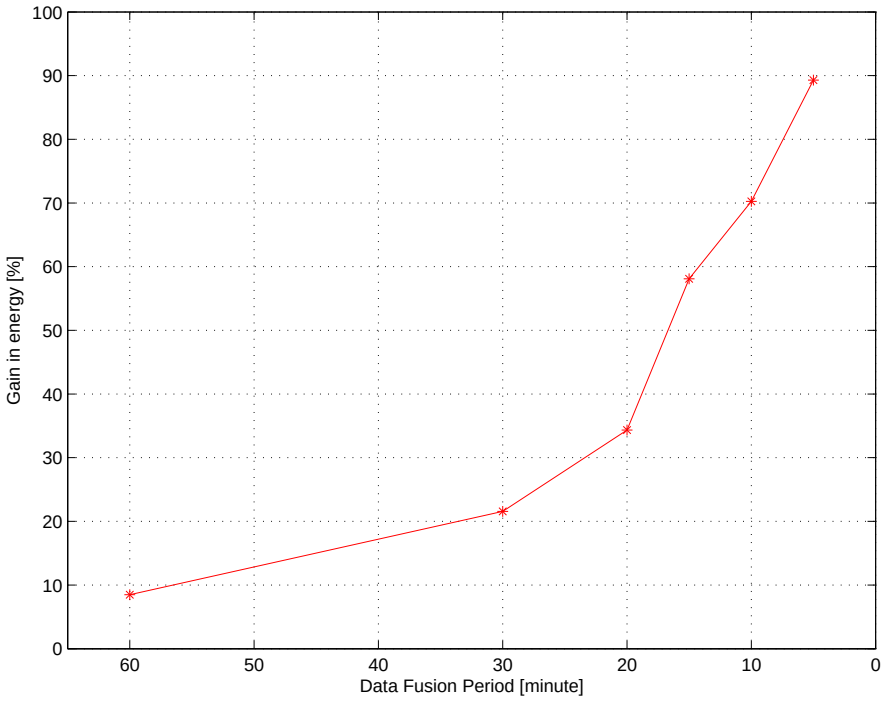


FIGURE 6.9: Gain in energy consumption

Contrary to the previous weighted methods that either calculate the weight of each evidence based on similarity distance to enhance credibility degree, or take into consideration the information volume to upgrade the uncertainty degree of evidence. DFIOT takes into consideration several parameters when calculating the mass function. DFIOT has several advantages: First, It improves the Deng entropy in uncertainty degree of evidence by giving more importance to sensors which have low uncertainty, i.e., more volume of information and less evidence distance vs. other sensors and by eliminating the evidence with the farthest distances when making decisions. It applies evidence distance in measuring conflict degree

and credibility. The second advantage that has been considered is the contextual parameters, which assure accuracy of the information affected by the resolution of the space. It focuses on the most recent information and avoids invalid information using the lifetime of sensed data. Another critical advantage of our method is that it helps to take the right decision when fusing the critical information. Examples of this include critical medical areas that need the lifetime of measurements to make a sensible decisions, in military applications that take the distance between entity and sensors as essential parameters, and in smart buildings that help to combine measurements and take optimal actions.

6.5 Conclusion

A novel data fusion method for the Internet of Things, we called DFIOT, has been proposed in this chapter. This method uses an adaptive weighted fusion algorithm based on D-S theory. The reliability of devices in the network and the conflicts between devices are considered in DFIOT. This by considering the information lifetime, the distance separating sensors and entities, reducing the computation and by using combination rules based on the Basic Probability Assignment, which allows to represent uncertain information or to quantify the similarity between two bodies of evidence. We compared the proposed solution with some state-of-the-art data fusion methods, including D-S, Murphy, Deng and Yuan, and using both benchmark data simulation and real dataset from a smart building testbed. Results show that DFIOT outperforms all the above mentioned methods in terms of reliability, accuracy and conflict management. The impact of this improvement from the application performance perspective has also been investigated, and the results show a gain of up to 90% in energy saving when using DFIOT.

Chapter 7

Conclusion and Future Directions

7.1 Conclusion

Over the past few decades, the world has seen the emergence of a new technology called the Internet of Things, which has grown rapidly and to this day it is still evolving. It has attracted the attention of many specialists in various fields, The IoT enables physical objects to see, hear, think and perform jobs by having them talk together, to share information and to coordinate decisions. The IoT transforms these objects from being traditional to smart by exploiting its underlying technologies such as ubiquitous and pervasive computing, embedded devices, communication technologies, sensor networks, Internet protocols and applications. Smart objects along with their supposed tasks constitute domain specific applications such as healthcare,transportation, agriculuture, smart cities, etc.

The interconnection of detection and actuation devices offering the ability to share platform information in a unified framework and services oriented architecture. This is achieved through large-scale detection, data processing and information representation using ubiquitous detection and even emerging technologies such as edge,fog, and cloud computing. exchanging data and information while reacting autonomously to the real/physical events of the world and influencing the execution of processes that trigger actions and create services with or without direct human intervention. IoT is carried out in three paradigms: a middleware, oriented objects (sensors) and oriented semantics (knowledge). The challenge is how to process heterogeneity and huge data, fusion with other data sources, to produce knowledgeable insight into data patterns for fast accurate decision making.

A highlight of the IoT is its pervasiveness in all areas of life. The IoT design must be pervasive, interoperable, consistent, reliable, efficient, and secure. Meanwhile, data that must be supported in the IoT can be multisource, heterogeneous, massive, redundant, inconsistent, or unreliable. Data fusion is an important tool that aim to combine measurements makes information more intelligent, decisive, sensible and precise which is coming from multiple sensors and sources. It can be helpful in handling the big data issues of IoT because we are fusing data from many sensors into more precise and accurate information. However, Data perceived by various sensors may be uncertain, inaccurate due to data loss or data source unreliability, which brings additional challenges for data fusion caused by data imperfection, data conflict, data ambiguity and inconsistency.

In this dissertation, we have dealt with these limitations by discussing and proposing solutions that ensure data processing in the emerging IoT.

On a side, we have proposed two efficient data processing frameworks to ensure real time data processing and reduce the network energy. The first framework integrate sensors and RFID technologies to enable sophisticated services in the emerging internet of things. With this architecture which based on edge computing, communication overheads can be significantly reduced and service in the cloud ensure real time data processing. The second solution is a platform to monitor and control smart objects in the Internet of Things (IoT). This is through IPv4/IPv6, and by fusing heterogeneity data and also combing IoT specific features and protocols such as CoAP, HTTP and WebSocket. The platform allows anomaly detection in IoT devices and real-time error reporting mechanisms. Moreover, the platform is designed as a standalone application, which targets at extending cloud connectivity to the network with fog computing. that allows heterogeneous resources, connectivity reliability and mobility, ensures security and contains services to enhance data fusion in application framework architecture.

On the other side, mathematical methods is indispensable when fusing data, it increases the quality of data, ensure reliability and scalability. These methods include data association, state estimation, and decision fusion. In order to ensure data credibility, reliability, reduce the data conflicts, and increase the data accuracy, we have proposed two methods. One in the feature data fusion and the other is decisional data fusion. We have proposed a new algorithm with new metric. The proposed method address the measurement uncertainty problem and make the particle filter robust to environmental change. Such method can be used in state estimation to fuse data and applied in smart environments and Internet

of things applications. This by estimate the kinematic parameters of the target. the aims of our proposal is to calculate the distance between probability densities which is described using Gaussian distribution and generate the optimal importance proposal distribution. The various estimation techniques are compared by computing the estimation root mean square error. The simulation results show that the proposed algorithm, ensures scalability and outperforms the standard particle filter, the improved particle filter based on KLD, and consensus based particle filter algorithm in high noise environment. Moreover, we have proposed a new efficient method is proposed, which emphasizes the importance of reducing the uncertainty of the measurements and conflicts in data fusion. It takes into consideration the contextual IoT parameters while reducing the degree of uncertainty. It also enhance the entropy of Deng, by selecting a sensor report which has a big information volume and well supported by the others sensors. This by exploring similarity between evidence, and hence enhances credibility. Results show that the proposed solution outperforms all the above mentioned methods in terms of reliability, accuracy, and conflict management.

7.2 Future Research Directions

The proposed frameworks and mathematical methods for data fusion open perspectives for the application in the emerging IoT. Integrating a virtual assistant to facilitate interaction with the system is a possible perspective. As the second perspective, the realized mathematical methods can be added in the proposed framework. The third perspective is the integration of machine learning to allow the system to learn habits of users and plan actions. The last perspective in our agenda is the integration of artificial intelligence algorithms to enable the system making autonomous decisions when necessary. A typical example of this (on which we are currently working) is the use of augmented and virtual reality based to develop mobile interfaces for IoT application system. The data fusion framework proposed in this chapter facilitates the development of these solutions.

Our Contributions

The following papers are the fruits of our work during the dissertation:

Journal papers

[1] Sahar Boulkaboul and Djamel Djenouri. DFIOT: Data Fusion for Internet Of Things. *J. Netw. Syst. Manag. Springer*, 28(4):11361160,2020. doi: 10.1007/s10922-020-09519-y. URL https://doi.org/10.1007/s10922-020-09519-y.

[2] Djamel Djenouri, Elmouatezbillah Karbab, Sahar Boulkaboul, and Antoine B.Bagula. Networked wireless sensors, active RFID, and handheld devices for moderncar park management: WSN, RFID, and mob devs for car park management. *Int.J.Handheld Comput. Res. IGI Global*, 6(3):45, 2015. doi: 10.4018/IJHCR.2015070103.
URL https://doi.org/10.4018/IJHCR.2015070103.

Conference papers

[3] Sahar Boulkaboul, Djamel Djenouri, Sadmi Bouhafs, and Mohand Ouamer Nait Belaid. Iot-DMCP: An IoT Data Management and Control Platform for smart cities. In *Proceedings of the 9th International Conference on Cloud Computing and Services Science, CLOSER 2019*, Heraklion, Crete, Greece, May 2-4, 2019, pages 578583. SciTePress, 2019. doi: 10.5220/0007861005780583. URL https://doi.org/10.5220/0007861005780583.

[4] Elmouatezbillah Karbab, Djamel Djenouri, Sahar Boulkaboul, and Antoine B. Bagula. Car park management with networked wireless sensors and active RFID. In *IEEE International Conference on Electro/Information Technology, EIT 2015,Dekalb, IL, USA*, May 21-23, 2015, pages 373378. IEEE, 2015. doi: 10.1109/EIT.2015.7293372. URL https://doi.org/10.1109/EIT.2015.7293372.

Bibliography

[1] Rafiullah Khan, Sarmad Khan, Rifaqat Zaheer, and Shahid Khan. Future internet: The internet of things architecture, possible applications and key challenges. pages 257–260, 12 2012. ISBN 978-1-4673-4946-8. doi: 10.1109/FIT.2012.53.

[2] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.*, 29(7):16451660, September 2013. ISSN 0167-739X. doi: 10.1016/j.future.2013.01.010.

[3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Comput. Netw.*, 54(15):27872805, October 2010. ISSN 1389-1286. doi: 10.1016/j.comnet.2010.05.010.

[4] F. Alam, R. Mehmood, I. Katib, N.N. Albogami, and A. Albeshri. Data fusion and iot for smart ubiquitous environments: A survey. *IEEE Access*, 5:9533–9554, 2017. doi: 10.1109/ACCESS.2017.2697839.

[5] Somayya Madakam, R Ramaswamy, and Siddharth Tripathi. Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3:164–173, 04 2015. doi: 10.4236/jcc.2015.35021.

[6] A. Juels. Rfid security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, 2006.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys and Tutorials*, 17(4):2347–2376, 2015. doi: 10.1109/COMST.2015.2444095.

[8] Vedat Coskun, Busra Ozdenizci, and Kerem Ok. A survey on near field communication (nfc) technology. *Wireless Personal Communications*, 71, 08 2013. doi: 10.1007/s11277-012-0935-5.

[9] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12):22922330, August 2008. ISSN 1389-1286. doi: 10.1016/j.comnet.2008.04.002.

[10] Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta, and Yim Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30:1655–1695, 05 2007. doi: 10.1016/j.comcom.2006.12.020.

[11] Neil Gershenfeld. *When things start to think*. Coronet Books Hodder & Stoughton, 1999. ISBN 978-0-8050-5874-1.

[12] Ruchi Garg and Sanjay Sharma. A study on need of adaptation layer in 6lowpan protocol stack. *International Journal of Wireless and Microwave Technologies*, 7:49–57, 05 2017. doi: 10.5815/ijwmt.2017.03.05.

[13] Roy Thomas Fielding and Richard N. Taylor. *Architectural Styles and the Design of Network-Based Software Architectures*. PhD thesis, 2000.

[14] G. Tanganelli, C. Vallati, and E. Mingozzi. Coapthon: Easy development of coap-based iot applications with python. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 63–68, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society. doi: 10.1109/WF-IoT.2015.7389028.

[15] Samir Chouali, Azzedine Boukerche, and Ahmed Mostefaoui. Towards a formal analysis of mqtt protocol in the context of communicating vehicles. In *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '17, pages 129–136, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5163-8. doi: 10.1145/3132062.3132079.

[16] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vzquez-Gallego, and Jess Alonso-Zrate. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud Computing*, 1(1), January 2015. doi: 10.5281/zenodo.51613.

[17] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997. doi: 10.1109/5.554205.

[18] Jens Bleiholder and Felix Naumann. Data fusion. *ACM Comput. Surv.*, 41 (1), January 2009. ISSN 0360-0300. doi: 10.1145/1456650.1456651.

[19] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28 – 44, 2013. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2011.08.001.

[20] H.B Mitchell. Multi-sensor data fusion: An introduction, 2007.

[21] Data fusion. https://algo-data.quora.com/Data-Fusion-an-overview-of-some-relevant-works, 2021.

[22] Hugh F. Durrant-Whyte. *Sensor Models and Multisensor Integration*, pages 73–89. Springer New York, New York, NY, 1990. ISBN 978-1-4613-8997-2. doi: 10.1007/978-1-4613-8997-2_7.

[23] Federico Castanedo. A review of data fusion techniques. *The Scientific World Journal*, 2013. doi: 10.1155/2013/704504.

[24] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014.

[25] Billy Pik Lik Lau, M. S. Hasala, Y. Zhou, N. Hassan, C. Yuen, Meng Zhang, and U. Tan. A survey of data fusion in smart city applications. *Inf. Fusion*, 52:357–374, 2019.

[26] Nour-Eddin El Faouzi, Henry Leung, and Ajeesh Kurian. Data fusion in intelligent transportation systems: Progress and challenges - a survey. *Inf. Fusion*, 12(1):410, January 2011. ISSN 1566-2535. doi: 10.1016/j.inffus.2010.06.001.

[27] Redowan Mahmud, Fernando Luiz Koch, and Rajkumar Buyya. Cloud-fog interoperability in iot-enabled healthcare solutions. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, ICDCN '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450363723. doi: 10.1145/3154273.3154347.

[28] Rustem Dautov, Salvatore Distefano, and Rajkumaar Buyya. Hierarchical data fusion for smart healthcare. *Journal of Big Data*, 6, 12 2019. doi: 10.1186/s40537-019-0183-6.

[29] Abder Rezak Benaskeur and Franois Rhaume. Adaptive data fusion and sensor management for military applications. *Aerospace Science and Technology*, 11(4):327 – 338, 2007. ISSN 1270-9638. COGIS '06.

[30] A. Sadeghi, C. Wachsmann, and M. Waidner. Security and privacy challenges in industrial internet of things. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2015. doi: 10.1145/2744769.2747942.

[31] M. S. Mekala and P. Viswanathan. A survey: Smart agriculture iot with cloud computing. In *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, pages 1–7, 2017. doi: 10.1109/ICMDCS.2017.8211551.

[32] Gang Sun, Victor Chang, Sheng-Uei Guan, Muthu Ramachandran, Jin Li, and Dan Liao. Big data and internet of thingsfusion for different services and its impacts. *Future Generation Computer Systems*, 86:1368–1370, 09 2018. doi: 10.1016/j.future.2018.05.022.

[33] Wenxiu Ding, Xuyang Jing, Zheng Yan, and Laurence Yang. A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion. *Information Fusion*, 51, December 2018. doi: 10.1016/j.inffus.2018.12.001.

[34] Yassine Himeur, Abdullah Alsalemi, Ayman Al-Kababji, Faycal Bensaali, and Abbes Amira. Data fusion strategies for energy efficiency in buildings: Overview, challenges and novel orientations. *Information Fusion*, 64:99 – 120, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.07.003.

[35] Rustem Dautov, Salvatore Distefano, and Rajkumaar Buyya. Hierarchical data fusion for smart healthcare. *Journal of Big Data*, 6, 12 2019. doi: 10.1186/s40537-019-0183-6.

[36] Nashreen Nesa and Indrajit Banerjee. Combining merkle hash tree and chaotic cryptography for secure data fusion in iot. In Marina L. Gavrilova, C. J. Kenneth Tan, Khalid Saeed, and Nabendu Chaki, editors, *Transactions on Computational Science*, pages 85–105, Berlin, Heidelberg, 2020. Springer Berlin Heidelberg. ISBN 978-3-662-61092-3.

[37] Ana Reyna, Cristian Martn, Jaime Chen, Enrique Soler, and Manuel Daz. On blockchain and its integration with iot. challenges and opportunities. *Future Generation Computer Systems*, 88:173 – 190, 2018. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2018.05.046.

[38] Klemen Kenda, Blaz Kazic, Erik Novak, and Dunja Mladeni. Streaming data fusion for the internet of things. *Sensors*, 19:1955, 04 2019. doi: 10.3390/s19081955.

[39] Rajiv Ranjan, Meisong Wang, Charith Perera, Prem Prakash Jayaraman, Miranda Zhang, Peter Strazdins, and R.K. Shyamsundar. City data fusion: Sensor data fusion in the internet of things. *Int. J. Distrib. Syst. Technol.*, 7(1):1536, January 2016. ISSN 1947-3532. doi: 10.4018/IJDST.2016010102.

[40] Jong-Woong Park, Sung-Han Sim, and Hyung-Jo Jung. Wireless displacement sensing system for bridges using multi-sensor fusion. *Smart Materials and Structures.*, 23(4):045022, mar 2014. doi: 10.1088/0964-1726/23/4/045022.

[41] S. Katoch, G. Muniraju, S. Rao, A. Spanias, P. Turaga, C. Tepedelenlioglu, M. Banavar, and D. Srinivasan. Shading prediction, fault detection, and consensus estimation for solar array control. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 217–222., 2018.

[42] Vincenzo Catania and Daniela Ventura. An approch for monitoring and smart planning of urban solid waste management using smart-m3 platform. volume 236, pages 24–31, 04 2014. ISBN 978-5-8088-0890-4. doi: 10.1109/FRUCT.2014.6872422.

[43] Feng Tian. An agri-food supply chain traceability system for china based on rfid and blockchain technology. pages 1–6, 06 2016. doi: 10.1109/ICSSSM.2016.7538424.

[44] Sergio Consoli, Diego Reforgiato Recupero, Misael Mongiovi, Valentina Presutti, Gianni Cataldi, and Wladimiro Patatu. An urban fault reporting and management platform for smart cities. In *Proceedings of the 24th International Conference on World Wide Web*, page 535540, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334730. doi: 10.1145/2740908.2743910.

[45] F. Ahmed and Y.E. Hawas. An integrated real-time traffic signal system for transit signal priority, incident detection and congestion management. *Transportation Research Part C: Emerging Technologies*, 60(C):52–76, 2015. doi: 10.1016/j.trc.2015.08.004.

[46] H. M. Hondori, M. Khademi, and Cristina V Lopes. Monitoring intake gestures using sensor fusion (microsoft kinect and inertial sensors) for smart

home tele-rehab setting. In *IEEE HIC 2012 Engineering in Medicine and Biology Society Conference on Healthcare Innovation*, Houston, TX, Nov 7-9 2012.

[47] S. Izumi and S. Azuma. Real-time pricing by data fusion on networks. *IEEE Transactions on Industrial Informatics*, 14(3):1175–1185, 2018.

[48] Soumya Kanti Datta and Christian Bonnet. An edge computing architecture integrating virtual IoT devices. In *GCCE 2017, IEEE 6th Global Conference on Consumer Electronics, October 24-27, 2017, Nagoya, Japan*, Nagoya, JAPON, 10 2017. doi: http://dx.doi.org/10.1109/GCCE.2017.8229253.

[49] Gabriel Mujica, Roberto Rodriguez-Zurrunero, Mark Richard Wilby, Jorge Portilla, Ana Beln Rodrguez Gonzlez, Alvaro Araujo, Teresa Riesgo, and Juan Jos Vinagre Daz. Edge and fog computing platform for data fusion of complex heterogeneous sensors. *Sensors*, 18(11), 2018. ISSN 1424-8220. doi: 10.3390/s18113630.

[50] Florian Heimgaertner, Stefan Hettich, Oliver Kohlbacher, and Michael Menth. Scaling home automation to public buildings: A distributed multiuser setup for openhab 2. In *GIoTS*, pages 1–6. IEEE, 2017.

[51] John Soldatos, Nikos Kefalakis, and Manfred Hauswirth. Openiot: Open source internet-of-things in the cloud. In *Interoperability and Open-Source Solutions for the Internet of Things*, pages 13–25, Cham, 2015. Springer International Publishing. ISBN 978-3-319-16546-2.

[52] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014.

[53] H. Lee, B. Lee, K. Park, and R. Elmasri. Fusion techniques for reliable information: A survey. *International Journal of Digital Content Technology and its Applications*, 4(2):74–88, 2010. doi: 10.4156/jdcta.vol4.issue2.9.

[54] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28 – 44, 2013. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2011.08.001.

[55] Y. Zheng. Methodologies for cross-domain data fusion: An overview. *IEEE Transactions on Big Data*, 1(1):16–34, March 2015. ISSN 2332-7790. doi: 10.1109/TBDATA.2015.2465959.

[56] Ivan Miguel Pires, Nuno M. Garcia, Nuno Pombo, and Francisco Flrez-Revuelta. From data acquisition to data fusion: A comprehensive review and a roadmap for the identification of activities of daily living using mobile devices. *Sensors*, 16(2), 2016. ISSN 1424-8220. doi: 10.3390/s16020184.

[57] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964.

[58] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. The probabilistic data association filter. *IEEE Control Systems Magazine*, 29(6):82–100, 2009. doi: 10.1109/MCS.2009.934469.

[59] Roger Higdon. *Multiple Hypothesis Testing*. Springer New York, New York, NY, 2013. ISBN 978-1-4419-9863-7. doi: 10.1007/978-1-4419-9863-7_1211.

[60] W. Sung, J. Chen, D. Huang, and Y. Ju. Multisensors realtime data fusion optimization for iot systems. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2299–2304, 2014.

[61] P. Jorge Escamilla-Ambrosio and N. Mort. Hybrid kalman filter-fuzzy logic adaptive multisensor data fusion architectures. *The 42th IEEE Conference on Decision and Control*, 5:5215–5220, January 2004. doi: 10.1109/CDC.2003.1272465.

[62] Jesse Read, Katrin Achutegui, and Joaquín Míguez. A distributed particle filter for nonlinear tracking in wireless sensor networks. *Signal Process.*, 98: 121–134, May 2014. ISSN 0165-1684. doi: 10.1016/j.sigpro.2013.11.020.

[63] Jeffrey K. Uhlmann. Covariance consistency methods for fault-tolerant distributed data fusion. *Inf. Fusion*, 4(3):201–215, 2003.

[64] Qing Da, Yang Yu, and Zhi-Hua Zhou. Learning with augmented class by exploiting unlabeled data. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, page 17601766. AAAI Press, 2014.

[65] Min Huang, Zhen Liu, and Yang Tao. Mechanical fault diagnosis and prediction in iot based on multi-source sensing data fusion. *Simulation Modelling Practice and Theory*, 102:101981, 2020. ISSN 1569-190X. doi: https://doi.org/10.1016/j.simpat.2019.101981. Special Issue on IoT, Cloud, Big Data and AI in Interdisciplinary Domains.

[66] Miguel ngel Lpez Medina, Macarena Espinilla, Cristiano Paggeti, and Javier Medina Quero. Activity recognition for iot devices using fuzzy spatio-temporal features as environmental sensor fusion. *Sensors*, 19(16), 2019. ISSN 1424-8220. doi: 10.3390/s19163512.

[67] D.S. Friedlander and S. Phoha. Semantic information fusion for coordinated signal processing in mobile sensor networks. *The International Journal of High Performance Computing Applications*, 16(3):235–241, 2002. doi: 10.1177/10943420020160030401.

[68] A. De Paola, P. Ferraro, S. Gaglio, G. L. Re, and S. K. Das. An adaptive bayesian system for context-aware data fusion in smart environments. *IEEE Transactions on Mobile Computing*, 16(6):1502–1515, 2017.

[69] N. Nesa and I. Banerjee. Iot-based sensor data fusion for occupancy sensing using dempstershafer evidence theory for smart buildings. *IEEE Internet of Things Journal*, 4(5):1563–1570, 2017.

[70] Ondrej Hlinka, Franz Hlawatsch, and Petar M. Djuric. Distributed particle filtering in agent networks: A survey, classification, and comparison. *IEEE Signal Processing Magazine*, 30(1):61–81, 2013. doi: 10.1109/MSP.2012.2219652.

[71] Tadesse Ghirmai. Distributed particle filter for target tracking: With reduced sensor communications. *Sensors*, 16(9), 2016. ISSN 1424-8220. doi: 10.3390/s16091454.

[72] Mark Coates. Distributed particle filters for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, IPSN '04, pages 99–107, New York, NY, USA, 2004. ACM. ISBN 1-58113-846-6. doi: 10.1145/984622.984637.

[73] Majdi Mansouri, Hazem N. Nounou, and Mohamed N. Nounou. Improved particle filtering for state and parameter estimation- CSTR model. pages 1–6, 2014. doi: 10.1109/SSD.2014.6808794.

[74] H. Q. Liu, H. C. So, F. K. W. Chan, and K. W. K. Lui. Distributed particle filter for target tracking in wireless network, Progress In Electromagnetics Research C, Vol. 11, 171182, 2009.

[75] Roi Yozevitch and Boaz Ben-Moshe. Advanced particle filter methods. In Javier Del Ser Lorente, editor, *Heuristics and Hyper-Heuristics*, chapter 5. IntechOpen, Rijeka, 2017. doi: 10.5772/intechopen.69236.

[76] Joelle Al Hage, Maan El Badaoui El Najjar, and Denis Pomorski. Multi-sensor fusion approach with fault detection and exclusion based on the kullback-leibler divergence: Application on collaborative multi-robot system. *Information Fusion*, 37:61–76, 01 2017. doi: 10.1016/j.inffus.2017.01.005.

[77] Jesse Read, Katrin Achutegui, and Joaquín Míguez. A distributed particle filter for nonlinear tracking in wireless sensor networks. *Signal Process.*, 98: 121–134, May 2014. ISSN 0165-1684. doi: 10.1016/j.sigpro.2013.11.020.

[78] Z. Yan, B. Zheng, and J. Cui. Distributed particle filter for target tracking in wireless sensor network. In *2006 14th European Signal Processing Conference*, pages 1–5, Sep. 2006.

[79] N. Ly-Tu, L. Mai, and T. Le-Tien. Kld-resampling with adjusted variance and gradient data-based particle filter applied to wireless sensor networks. In *2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pages 229–234, Sep. 2015. doi: 10.1109/NICS.2015.7302197.

[80] Yongli Wang and Jiangbo Qian. Measuring the uncertainty of rfid data based on particle filter and particle swarm optimization. *Wirel. Netw.*, 18 (3):307–318, April 2012. ISSN 1022-0038. doi: 10.1007/s11276-011-0401-4.

[81] Xiaofan Li, Yubin Zhao, Sha Zhang, and Xiaopeng Fan. Adaptive particle filter for nonparametric estimation with measurement uncertainty in wireless sensor networks. *Sensors*, 16(6), 2016. ISSN 1424-8220. doi: 10.3390/s16060786.

[82] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Efficient multiple importance sampling estimators. *IEEE Signal Processing Letters*, 22(10): 1757–1761, Oct 2015. doi: 10.1109/LSP.2015.2432078.

[83] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599623, Mar 2016. ISSN 1573-1375. doi: 10.1007/s11222-016-9642-5.

[84] D. stebay, M. Coates, and M. Rabbat. Distributed auxiliary particle filters using selective gossip. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3296–3299, May 2011. doi: 10.1109/ICASSP.2011.5946726.

[85] Márk Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. Gossip-based peer sampling. *ACM Trans. Comput. Syst.*, 25(3), August 2007. ISSN 0734-2071. doi: 10.1145/1275517.1275520.

[86] Vladimir Savic, Henk Wymeersch, and Santiago Zazo. Belief consensus algorithms for fast distributed target tracking in wireless sensor networks. *Signal Process.*, 95:149–160, February 2014. ISSN 0165-1684. doi: 10.1016/j.sigpro.2013.09.005.

[87] Jiwen W. Guan and D. A. Bell. *Evidence Theory and Its Applications*. Elsevier Science Inc., USA, 1992. ISBN 0444896414.

[88] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[89] Arthur P. Dempster. *Upper and Lower Probabilities Induced by a Multivalued Mapping*, pages 57–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-44792-4. doi: 10.1007/978-3-540-44792-4_3.

[90] Tazid Ali, Palash Dutta, and Hrishikesh Boruah. A new combination rule for conflict problem of dempster-shafer evidence theory. *International Journal of Energy, Information and Communications*, 3, 03 2012.

[91] Sylvie le hgarat, Isabelle Bloch, and Daniel Vidal-Madjar. Application of dempster-shafer evidence theory to unsupervised classification in multisource remote sensing. *Geoscience and Remote Sensing, IEEE Transactions on*, 35: 1018 – 1031, 08 1997. doi: 10.1109/36.602544.

[92] J.R. Boston. A signal detection system based on dempster-shafer theory and comparison to fuzzy detection. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30:45 – 51, 03 2000. doi: 10.1109/5326.827453.

[93] Yibing Li, Jie Chen, and Yun Lin. An efficient combination method of conflict evidences. *International Journal of Hybrid Information Technology*, 8:299–306, 12 2015. doi: 10.14257/ijhit.2015.8.12.22.

[94] Ronald Yager. Decision making using minimization of regret. *Int. J. Approx. Reasoning*, 36:109–128, 06 2004. doi: 10.1016/j.ijar.2003.10.003.

[95] R. R. Yager and D. P. Filev. Including probabilistic uncertainty in fuzzy logic controller modeling using dempster-shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(8):1221–1230, 1995.

[96] P. Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5): 447–458, 1990.

[97] Didler Dubois and Henri Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4(3):244–264, 1988. doi: 10.1111/j.1467-8640.1988.tb00279.x.

[98] Catherine K. Murphy. Combining belief functions when evidence conflicts. *Decision Support Systems*, 29(1):1 – 9, 2000. ISSN 0167-9236. doi: https://doi.org/10.1016/S0167-9236(99)00084-6.

[99] Anne-Laure Jousselme, Dominic Grenier, and loi Boss. A new distance between two bodies of evidence. *Information Fusion*, 2(2):91 – 101, 2001. ISSN 1566-2535. doi: https://doi.org/10.1016/S1566-2535(01)00026-4.

[100] Deng Yong, Shi WenKang, Zhu ZhenFu, and Liu Qi. Combining belief functions based on distance of evidence. *Decision Support Systems*, 38(3):489 – 493, 2004. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2004.04.015.

[101] Zhenjiang Zhang, Tonghuan Liu, Dong Chen, and Wenyu Zhang. Novel algorithm for identifying and fusing conflicting data in wireless sensor networks. *Sensors*, 14(6):95629581, May 2014. ISSN 1424-8220. doi: 10.3390/s140609562.

[102] Peiyi Zhu, Benlian Xu, and Baoguo Xu. An improved particle swarm optimization for uncertain information fusion. In Ying Tan, Yuhui Shi, Yi Chai, and Guoyin Wang, editors, *Advances in Swarm Intelligence*, pages 494–501, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21524-7.

[103] Shilpa Gite and Himanshu Agrawal. On context awareness for multisensor data fusion in iot. In Suresh Chandra Satapathy, K. Srujan Raju, Jyotsna Kumar Mandal, and Vikrant Bhateja, editors, *Proceedings of the Second International Conference on Computer and Communication Technologies*, pages 85–93, New Delhi, 2016. Springer India. ISBN 978-81-322-2526-3.

[104] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, 2014.

[105] Zartasha Baloch, Faisal Karim Shaikh, and Mukhtiar Ali Unar. A context-aware data fusion approach for health-iot. *International Journal of Information Technology*, 10:241–245, 2018.

[106] Yong Deng. Deng entropy: a generalized shannon entropy to measure uncertainty. *viXra*, 01 2015. doi: 10.5281/zenodo.32211.

[107] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[108] Tzu-Chao Lin. Improving ds evidence theory for data fusion system. 08 2008.

[109] Kaijuan Yuan, Fuyuan Xiao, Liguo Fei, Bingyi Kang, and Yong Deng. Conflict management based on belief function entropy in sensor fusion. *SpringerPlus*, 5, 12 2016. doi: 10.1186/s40064-016-2205-6.

[110] Antoine Bigomokero Bagula, Djamel Djenouri, and Elmouatezbillah Karbab. On the relevance of using interference and service differentiation routing in the internet-of-things. In *Internet of Things, Smart Spaces, and Next Generation Networking*, pages 25–35, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40316-3.

[111] V. W. s. Tang, Y. Zheng, and J. Cao. An intelligent car park management system based on wireless sensor networks. In *2006 First International Symposium on Pervasive Computing and Applications*, pages 65–70, 2006. doi: 10.1109/SPCA.2006.297498.

[112] Jonathan Benson, Tony O'Donovan, Padraig O'Sullivan, Utz Roedig, Cormac Sreenan, John Barton, Aoife Murphy, and Brendan O'Flynn. Car-park management using wireless sensor networks. In *The 31st Annual IEEE Conference on Local Computer Networks, Tampa, Florida, USA*, pages 588–595, 01 2006.

[113] Bi, Yan-zhong, Sun, Li-min, Zhu, Hongsong, Yan, Ting-Xin, Luo, and Zheng-jun. A parking management system based on wireless sensor network. *Acta Automatica Sinica*, 32:968–977, 2006.

[114] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva. Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Communications Surveys and Tutorials*, 17(3):1294–1312, 2015. doi: 10.1109/COMST.2015.2388550.

[115] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, LCN '04, pages 455–462, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2260-2. doi: 10.1109/LCN.2004.38.

[116] Mališa Vučinić, Bernard Tourancheau, Franck Rousseau, Andrzej Duda, Laurent Damon, and Roberto Guizzetti. Oscar. *Ad Hoc Netw.*, 32(C):3–16, September 2015. ISSN 1570-8705. doi: 10.1016/j.adhoc.2014.12.005.

[117] Matthias Kovatsch, Martin Lanter, and Zach Shelby. Californium: Scalable cloud services for the internet of things with coap. In *4th International Conference on the Internet of Things, IOT 2014, Cambridge, MA, USA, October 6-8, 2014*, pages 1–6, 2014. doi: 10.1109/IOT.2014.7030106.

[118] Robert B. Miller. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 267–277, New York, NY, USA, 1968. ACM. doi: 10.1145/1476589.1476628.

[119] Philippe Smets. Belief functions on real numbers. *International Journal of Approximate Reasoning*, 40(3):181 – 223, 2005. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2005.04.001.

[120] B Ristic and Ph. Smets. Belief function theory on the continuous space with an application to model based classification. In Bernadette Bouchon-Meunier, Giulianella Coletti, and Ronald R. Yager, editors, *Modern Information Processing*, pages 11 – 24. Elsevier Science, Amsterdam, 2006. ISBN 978-0-444-52075-3.

[121] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. . Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, Feb 2002. ISSN 1053-587X. doi: 10.1109/78.978396.

[122] Holly Moore. *MATLAB for Engineers. Prentice Hall Press.* 2014.

Abstract : The emergence IoT is rapidly gaining ground in our modern society, aiming to improve the quality of life by connecting many smart devices, technologies and applications, for the purpose of exchanging data over the Internet. IoT devices will generate huge volumes of data in a rapid period of time and therefore require scalable solutions for dynamic and real-time processing of the generated data. Such solutions should provide a high level of accurate and reliable data for decision making. This requires data fusion, which is an efficient way for optimal use of a huge volume of data from multiple sources. We consider in this thesis the integration of IoT with edge, fog and cloud computing, the efficiency of data processing and fusion in terms of credibility, reliability, conflict, latency, and we propose several solutions. The first concerns the efficient processing of data in edge computing, which enables sophisticated services. The second approach is a hybrid computing-based IoT data management and control platform that enables heterogeneous resources, reliable connectivity and mobility, provides security, and contains services to merge data. heterogeneous. Numerical analysis and simulation results show that the proposed solutions allow significant savings in terms of energy consumption and reduction of lead times. The thesis also considers the state estimation in the average level of data fusion. We provide an improved distributed particulate filter algorithm to process target tracking in wireless sensor networks. It increases the estimation accuracy of the particulate filter, improves the efficiency of particle sampling, and improves the estimation performance. The simulation and numerical analysis results show the superiority of the proposed approach in terms of root mean square error and scalability. We have studied the problem of data fusion at the decision-making level. Reliability and conflicts are taken into account in our method by considering the information lifetime, the distance between sensors and features, reducing the computation and using combination rules based on the base probability assignment . This makes it possible to represent uncertain information or to quantify the similarity between two bodies of evidence. We compared the proposed solution with state-of-the-art data fusion methods, and using both benchmark data simulation and an actual data set from an intelligent building testbed. The results show that our solution outperforms methods in terms of reliability, accuracy and conflict handling.
Keywords : Internet of Things; Edge Computing; Fog Computing; Cloud Computing; Dempster-Shafer theory; Confict management; Energy consumption; Basic Probability Assignment; Similarity distance; Weighted evidences.

Résumé : L'émergence de l'IoT gagne rapidement du terrain dans notre société moderne, visant à améliorer la qualité de vie en connectant de nombreux appareils, technologies et applications intelligents, dans le but d'échanger des données sur Internet. Les appareils IoT généreront d'énormes volumes de données dans un laps de temps rapide et nécessiteront donc des solutions évolutives pour le traitement dynamique et en temps réel des données générées. De telles solutions devraient fournir un niveau élevé de données précises et fiables pour la prise de décision.Cela nécessite une fusion données, qui est un moyen efficace pour une utilisation optimale d'un volume énorme de données provenant de sources multiples. Nous considérons dans cette thèse l'intégration de l'IoT à edge, fog et au cloud computing, l'efficacité du traitement et de la fusion des données en termes de crédibilité, de fiabilité, de conflit, de latence, et nous proposons plusieurs solutions. La première concerne le traitement efficace des données dans l'edge computing, qui permet des services sophistiqués. La deuxième approche est une plate-forme de gestion et de contrôle des données de l'IoT basée sur le hybride computing qui permet des ressources hétérogènes, la fiabilité de la connectivité et la mobilité, assure la sécurité et contient des services pour fusionner les données dhétérogéne. L'analyse numérique et les résultats de simulation montrent que les solutions proposées permettent des économies significatives en termes de consommation d'énergie et de réduction des délais. La thèse considère également l'estimation d'état dans le niveau moyen de fusion de données. Nous proposons un algorithme de filtre à particules distribué amélioré pour traiter le suivi de cible dans les réseaux de capteurs sans fil. Il augmente la précision d'estimation du filtre à particules, améliore l'efficacité de l'échantillonnage des particules et améliore les performances d'estimation. Les résultats de simulation et d'analyse numérique montrent la supériorité de l'approche proposée en termes d'erreur quadratique moyenne et d'évolutivité. Nous avons étudié le problème de la fusion de données au niveau décisionnel. La fiabilité et les conflits sont pris en compte dans notre méthode en considérant la durée de vie des informations, la distance séparant les capteurs et les entités, en réduisant le calcul et en utilisant des règles de combinaison basées sur l'affectation de probabilité de base. Cela permet de représenter des informations incertaines ou de quantifier la similitude entre deux corpus de preuves. Nous avons comparé la solution proposée avec des méthodes de fusion de données de pointe, et en utilisant à la fois une simulation de données de référence et un ensemble de données réel à partir d'un testbed de bâtiment intelligent. Les résultats montrent que notre solution surpasse toutes les méthodes en termes de fiabilité, de précision et de gestion des conflits.
Mots-clés : Internet des objets ; Edge Computing ; Fog Computing ; Cloud computing; théorie de Dempster-Shafer; Gestion de conflit ; consommation d'énergie; Affectation de probabilité de base; distance de similarité.

ملخص: يكتسب ظهور إنترنت الأشياء تقدمًا سريعًا في مجتمعنا الحديث ، بهدف تحسين نوعية الحياة من خلال ربط العديد من الأجهزة والتقنيات والتطبيقات الذكية ، بغرض تبادل البيانات عبر الإنترنت. تولد أجهزة إنترنت الأشياء أحجامًا ضخمة من البيانات في فترة زمنية سريعة ، وبالتالي تتطلب حلولًا قابلة للتطوير للمعالجة الديناميكية وفي الوقت الفعلي للبيانات التي تم إنشاؤها. يجب أن توفر مثل هذه الحلول مستوى عالٍ من البيانات الدقيقة والموثوقة لاتخاذ القرار. وهذا يتطلب دمج البيانات ، وهي طريقة فعالة للاستخدام الأمثل لحجم ضخم من البيانات من مصادر متعددة. نعتبر في هذه الأطروحة تكامل إنترنت الأشياء مع الحافة ، والحوسبة السحابية والضبابية ، وكفاءة معالجة البيانات ودمجها من حيث المصداقية ، والموثوقية ، والتعارضات ، والكمون ، ونقترح العديد من الحلول. الأول يتعلق بالمعالجة الفعالة للبيانات في الحوسبة المتطورة ، والتي تتيح خدمات متطورة. أما الأسلوب الثاني فهو عبارة عن نظام أساسي لإدارة بيانات إنترنت الأشياء والتحكم فيها يعتمد على الحوسبة ويتيح موارد غير متجانسة واتصال وتنقل موثوق به ويوفر الأمان ويحتوي على خدمات لدمج البيانات غير المتجانسة. تظهر نتائج التحليل العددي والمحاكاة أن الحلول المقترحة تسمح بتوفير كبير من حيث استهلاك الطاقة وتقليل المهل الزمنية. تتناول الأطروحة أيضًا تقدير الحالة في المستوى المتوسط لدمج البيانات. نقدم خوارزمية مرشح الجسيمات الموزعة المحسنة لمعالجة تتبع الهدف في شبكات الاستشعار اللاسلكية. يزيد من دقة تقدير مرشح الجسيمات ، ويحسن كفاءة أخذ عينات الجسيمات ، ويحسن أداء التقدير. تظهر نتائج المحاكاة والتحليل العددي تفوق النهج المقترح من حيث جذر متوسط الخطأ التربيعي وقابلية التوسع. لقد درسنا مشكلة دمج البيانات على مستوى إتخاذ القرار. تؤخذ الموثوقية والتعارضات في الاعتبار في طريقتنا من خلال النظر في وقت المعلومات ، والمسافة بين المستشعرات والميزات ، وتقليل الحساب واستخدام قواعد المجموعة بناءً على تعيين الاحتمال الأساسي. هذا يجعل من الممكن تمثيل معلومات غير مؤكدة أو تحديد التشابه بين مجموعتين من الأدلة. قارنا الحل المقترح بأحدث طرق دمج البيانات ، وباستخدام كل من محاكاة البيانات المعيارية ومجموعة البيانات الفعلية من اختبار بناء ذكي. تظهر النتائج أن حلنا يتفوق أداءا على الأساليب المدروسة من حيث الموثوقية والدقة ومعالجة النزاعات.
الكلمات المفتاحية: انترنت الأشياء؛ حوسبة الحافة حوسبة الضباب؛ حوسبة سحابية؛ نظرية ديمبستر شافر؛ إدارة الصراع؛ استهلاك الطاقة؛ الاحتمالية الأساسية ؛ مسافة التشابه ؛الأدلة المرجحة.