People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University 8 Mai 1945 Guelma
Faculty of Mathematics, Computer Science, and Sciences of Matter

DISSERTATION

In Partial Fulfillment of the Requirements for The **Doctorate** degree in 3rd cycle

Field: Computer Science

Specialty: Computer Science Systems

Presented by: : HADJADJ Abdelhalim

Entitled

An Internet of Things Framework Based on the Semantic Web and Big Data

Defended: 10/10/2023. Before, the examination board composed of:

Pr.	Yacine Lafifi	Professor	Univ. of 8 Mai 1945, Guelma	Chairman
Dr.	Khaled Halimi	Senior Lecturer	Univ. of 8 Mai 1945, Guelma	Supervisor
Pr.	Zineddine kouahla	Professor	Univ. of 8 Mai 1945, Guelma	A-Supervisor
Pr.	Hassina Seridi-Bouchelaghem	Professor	Univ. of Badji Mokhtar, Annaba	Examiner
Pr.	Brahim Farou	Professor	Univ. of 8 Mai 1945, Guelma	Examiner
Dr.	Karima Benhameza	Senior Lecturer	Univ. of 8 Mai 1945, Guelma	Examiner

I would like to dedicate this thesis to:

My parents,

For your unconditional love, unwavering support, and endless sacrifices. Your confidence in me and your constant encouragement has been the driving force that has allowed me to persevere in this academic journey. Thank you for everything you have done for me.

My wife,

For your infinite patience and understanding. Your unfailing support and presence by my side have been my anchor in difficult times and have been a constant source of inspiration and motivation for me. I am grateful to have you in my life.

To my children, Mohamed and Elbara, *who are the source of my greatest inspiration and motivation.*

To the soul of my uncle,**Abdelaziz Oughidni**, *for all of his encouragement and support and the profound impact he has given on my life.*

To my brothers and friends, Abderhmen K, Chokri F, Mohamed S, Walid O, and Abdelhak S, members of the Labstic and the computer science department, *for your moral support and encouragement.*

To ***all*** *those who have contributed to my research.*

Remerciement

After a long period of work and effort, I finally reach the end of this adventure with the writing of this thesis. Achieving this thesis was a difficult task and I could not do it without the help of Allah and the guidance of my supervisors, colleagues, family, and friends.

First of all, I would like to express my sincere thanks to ALLAH, Lord of the worlds, who gave me the opportunity to perform this work. I am so grateful for His guidance, mercy, and blessing along the way.

I would also like to thank my supervisors Halimi Khaled and Zinedinne Kouahla for their support, guidance, and patience throughout this project. Your experience, advice, and guidance contributed greatly to the success of this research.

I would like to express my gratitude to all the members of the examination board Yacine Lafifi, Hassina Seridi-Bouchelaghem, Brahim Farou, Karima Benhamza for the honor they give me by accepting to be the rapporteurs of this thesis.

Finally, I am grateful to the Labstic laboratory member and their manager's contribution to the completion of this thesis.

Abstract

The Internet of Things (IoT) operates across various domains, such as healthcare, with the aim of enhancing performance through the remote and real-time collection of data. This technology facilitates the monitoring of patients' health status by measuring their vital signs. However, to fully exploit the potential of the IoT and take advantage of all the opportunities it offers, it is necessary to solve the problems of heterogeneity and the lack of interoperability.

The semantic web of things is a solution developed by researchers to solve these problems by integrating the Semantic Web and the Internet of Things. This framework is based on semantic technologies such as RDF, RDFS, OWL and ontology. However, using various ontology development methods leads to creating specific ontologies that cause alignment and sharing problems, especially when the domain represented by this ontology is interconnected with other domains, such as healthcare related to transportation, education, etc. This framework also suffers from a lack of capacity to deal with the vague and imprecise data that characterize the health domain. This state can lead to inaccurate processing and incorrect results, which is unacceptable in this domain where accuracy is crucial for decision-making. As this framework is designed to collect data from numerous IoT devices, this leads to a large amount of data in RDF format, also known as BIG RDF data. To process this data efficiently, an efficient storage and retrieval method is necessary. In addition, the framework should enable easy incorporation of new IoT devices as well as ensure real-time processing.

In this thesis, to overcome these problems, a semantic web of things framework has been developed. The development of this framework is accomplished through three contributions. Firstly, an ontology has been developed, which is the basis of the framework and has been developed using the neon methodology. The ontology is an extension of the standard SAREF ontology that supports alignment and sharing. The ontology allows a representation of their integration of domain health and public transport. Secondly, to address the problem of vague and imprecise data, another more advanced ontology that can deal with vague and imprecise data in health data was developed. The ontology has represented a transformation of our SAREF ontology extension for COVID-19 to fuzzy ontology, which helps the framework for more accurate and reliable decision-making. Third, adding the cluster and indexing layers to the framework can process the BIG RDF data by grouping these outputs into clusters and narrowing the search space. Finally, systems based on an IoT architecture and using early warning systems such as MEWS and NEWS2 have been developed to validate these contributions. These systems allow the determination of the patient's health status, which allows the provision of appropriate health services. The results obtained with the developed system were considered very promising and encouraging.

Keywords: Internet of things, Healthcare, semantic web, fuzzy logic, clustering, index-

ملخص

يعمل إنترنت الأشياء على تحسين الأداء في العديد من المجالات ، بما في ذلك المجال الصحي ، حيث يسمح بجمع و تشارك البيانات عن بُعد وفي الوقت الفعلي .فهو يسمح بمراقبة الحالة الصحية للمرضى من خلال قياس إشاراتهم الحيوية. ومع ذلك ، للاستفادة الكاملة من إمكانات إنترنت الأشياء ، من الضروري حل مشاكل عدم التجانس بين البيانات والحاجة إلى قابلية التشغيل البيني.

تعد الشبكة الدلالية للأشياء حلاً طوره الباحثون لحل هذه المشكلات من خلال دمج الويب الدلالي وإنترنت الأشياء. يعتمد هذا الإطار على التقنيات الدلالية مثل RDF و RDFS و OWL وعلم الوجود. ومع ذلك ، يؤدي استخدام أساليب تطويرا لأنطولوجيا المختلفة إلى إنشاء أنطولوجيا محددة تسبب مشكلات التنسيق والمشاركة ، خاصة عندما يكون المجال الذي يمثله علم الوجود مترابطاًا مع مجالات أخرى ، مثل الرعاية الصحية تتعلق بمجالات النقل والتعليم وما إلى ذلك.

إضافة إلى ذلك، يعاني هذا الإطار من نقص في القدرة على التعامل مع البيانات الغامضة وغير الدقيقة التي تميز العديد من المجالات كالمجال الصحي. يمكن أن تؤدي هذه الحالة إلى معالجة غير دقيقة ونتائج غير صحيحة، وهو أمر غير مقبول حيث تكون الدقّة ضرورية لاتخاذ القرارات. و نظراً لأن هذا الإطار مصمم لمعالجة البيانات المأخوذة من العديد من أجهزة إنترنت الأشياء، فإن هذا يؤدي إلى إنتاج كمية كبيرة من البيانات بتنسيق RDF ، والمعروفة أيضاًا باسم BIG RDF data . ليتمكن هذا الإطار من معالجة هذه البيانات بكفاءة، فمن الضروري أن يعتمد على طريقة تخزين واسترجاع فعالة. فضلاً على ذلك، يجب أن يوافق هذا الإطار عملية الدمج السهل لأجهزة إنترنت الأشياء الجديدة و المعالجة في الوقت الفعلي.

للتغلب على هذه المشاكل، في هذه الأطروحة، تم تطوير شبكة دلالية للأشياء.حيث تم تطوير هذا الإطار من خلال ثلاث مساهمات. أولاً، تم تطوير علم الوجود ، الذي يعدّ أساس هذا الإطار وقد تم تطويره باستخدام منهجية نيون. يعد علم الوجود المطور امتداد لعلم الوجود المعياري SAREF و الذي يدعم التنسيق والمشاركة. كما سمح علم الوجود المطور بتمثيل دمج كل من المجالين الصحي والنقل العام. ثانياًا ، لمعالجة مشكلة البيانات الغامضة وغير الدقيقة ، تم تطوير علم وجود يمكنه التعامل مع البيانات الغامضة وغير الدقيقة. لقد تم تطويره من خلال تحويل امتداد علم الوجود SAREF وفق منهجية Ikarus إلى علم الوجود يستطيع التعامل مع البيانات الغامضة كالتي يحويها مجال كوفيد١٩ ، مما يساعد الإطار على اتخاذ قرارات أكثر دِقّة وموثوقيه.

ثالثا ، من خلال إضافة كل من طبقت التجميع و طبقة الفهرسة إلى إطار العمل، تمكن الإطار من معالجة مشكلة التخزين و البحث لبياناته على شكل BIG RDF عن طريق تجميعها في مجموعات وتضييق مساحة البحث. أخيراً، للتحقق من صحة هذه المقترحات، تم تطويرا أنظمة ترتكز على هندسة أنترنت الأشياء و تستخدم أنظمة الإنذار المبكر مثل MEWS و NEWS2. سمحت هذه الأنظمة المطورة بتحديد الحالة الصحية للمريض و تقديم الخِدْمَات الصحية المناسبة. تعدّ النتائج المتحصل عليها واعدة ومشجعة للغاية.

الكلمات المفتاحية: إنترنت الأشياء ، الرعاية الصحية ، الويب الدلالي ، المنطق الضبابي ، التجميع ، الفهرسة.

Résumé

L'internet des objets(Ido) est utilisé dans de nombreux domaines, notamment dans le domaine de la santé, pour améliorer les performances en permettant la collecte de données à distance et en temps réel. Cependant, pour exploiter pleinement le potentiel de l'IdO et profiter de toutes les opportunités qu'il offre, il est nécessaire de résoudre les problèmes d'hétérogénéité et le besoin d'interopérabilité. Des chercheurs ont développé le Web sémantique des objets comme une solution visant à résoudre ces problèmes en intégrant la sémantique dans le contexte de l'Internet des objets. Cette solution est basée sur les sémantiques technologies telles que RDF, RDFS, et OWL qui représente les ontologies. Cependant, l'utilisation de différentes méthodes de développement d'ontologies et la spécificité de ces ontologies peuvent poser des problèmes d'alignement et de partage, notamment dans le domaine de la santé, qui est intégré avec d'autres domaines tels que le transport et l'éducation, etc. Ce framework souffre aussi d'un manque de capacité à traiter les données vagues et imprécises qui caractérisent le domaine de la santé. Cet état peut conduire à des traitements imprécis et des erreurs de résultats. Étant donné que ce framework est conçu pour collecter des données provenant d'un grand nombre de dispositifs IoT, cela entraîne une quantité importante de données en format RDF, également appelé BIG RDF data. Pour traiter efficacement ces données, il est nécessaire de disposer d'une méthode de stockage et de recherche efficace. Le framework aussi doit support l'ajout de nouveaux IoD dispositifs ainsi que garantir le traitement en temps réel.

Dans cette thèse, pour remédier à ces problèmes, un cadre sémantique web des objets a été développé. Le développement de ce cadre est réalisé à travers trois contributions. Premièrement, une ontologie a été développée, qui constitue la base du cadre et a été développée à l'aide de la méthodologie néon.L'ontologie est une extension de standard ontologie nommé SAREF ontologie qui est supporté l'alignement et le partage. L'ontologie permettre une représentation à l'intégration de domaine santé et transport publique. Deuxièmement, pour résoudre le problème des données vagues et imprécises, une autre ontologie plus performante, capable de traiter les données vagues et imprécises dans les données de santé, a été développée. La nouvelle ontologie a représenté une transformation de notre extension de l'ontologie SAREF pour COVID-19 en ontologie floue, ce qui aide le cadre à prendre des décisions plus précises et plus fiables. Troisièmement, par l'ajout des couches de regroupement et d'indexation au cadre, il est possible de traiter les données BIG RDF en regroupant ces résultats en clusters et en réduisant l'espace de recherche. Enfin, des systèmes basés sur une architecture IoT et utilisant des systèmes d'alerte précoce tels que MEWS et NEWS2 ont été développés pour valider ces contributions. Les résultats obtenus grâce à ces approches ont été considérés comme très prometteurs et encourageants.

Mots Clée : Internet des objets, santé, web sémantique, logique floue, regroupement, indexation.

Contents

List of Figures

List of Tables

General Introduction

The Internet of Things (IoT) is an emerging technology based on the interconnection between objects over the internet, which makes these objects more intelligent to help humans in the activities of daily life. The internet of things provides many capabilities that are represented by the improvement of operational efficiency and the automation of repetitive tasks, which increases productivity and reduces costs. It also helps enterprises to develop new products and services, which led to improved quality of life. With these capabilities, this technology has the potential to develop many fields, such as agriculture, industry, environment, and many more.

The health system aims to safeguard and improve human health by providing access to high-quality health services, preventing disease, and offering the necessary resources for diagnosis and treatment. However, the high number of diseases and their high-risk lead to death, in addition to their spreading speed, particularly contagious diseases, which have resulted in pandemics and the deaths of millions of people. Furthermore, hospital overcrowding, availability of health services and high costs, the impact of pollution, poverty, and famine in the Third World, and other problems still remain significant barriers to effective healthcare delivery.

To overcome these problems, the Internet of Things is an effective solution. It enables a quantum leap in healthcare services by providing them outside of clinics through applications that enable the exchange of health information and provide remote medical monitoring. It saves a large proportion of patients who need to monitor their health on a daily basis from going to the hospital and allows their doctor to be informed in a timely manner of the evolution of their health condition. Furthermore, it also enables rapid intervention in case of complications and reduces effort, costs, and time. Despite these significant contributions to improving the health sector, this technology includes problems that led to a decrease in its efficiency, such as heterogeneity and lack of interoperability.

Motivation

In recent years, the healthcare domain has gained significant importance, with a focus on improving the quality of health services and preventing disease. Governments and countries have taken significant efforts to enhance healthcare resources and infrastructure. On the other hand, the internet of things domain has proliferated in all domains and has witnessed a large increase in the number of devices that are characterized by their small size and low cost. This means the integration of this technology in the healthcare domain is an ideal solution to enhance and improve the performance of the healthcare system. The IoT can play an important role in healthcare. It can collect data from IoT devices and then process or share it with healthcare system actors such as doctors, nurses, hospitals, etc. It enables remote patient monitoring, improved patient experience, enhanced drug management, etc. Furthermore, the internet of things can benefit from the semantic web to tackle the heterogeneity and lack of interoperability problem. The integration of IoT with the semantic web is an adequate solution that enables the structure of the data consistently and understandably, allowing for more efficient use of the data. It facilitates the exchange of patient data between different healthcare providers, as well as the integration of medical devices and applications. The integration of the semantic web and the internet of things has led to the emergence of the semantic web of things. The purpose of the semantic web of things is to solve the issues with data heterogeneity and provide a better comprehension and use of data, as well as the development of new services and applications.

In addition to these benefits, the improved processing of its IoT data and storage of its output and the capacity to retrieve this data very quickly will lead to great efficiency that will satisfy the requirements of the healthcare domain and contribute to the advancement of the internet of things domain.

Problematic

As the Internet of Things becomes more widely used, there is a need for devices and applications to be interoperable and able to share data seamlessly across the network. Interoperability allows quick and easy access to data, as well as the integration of new IoT devices and applications, which improves the efficiency and usability of IoT technology. However, the wide variety of hardware, software, platforms, and communications protocols that comprise the Internet of Things are unable to communicate effectively with one another and exchange IoT data, making it difficult to create an interoperable IoT system. This state is causing the internet of things to lose its economic potential by 40% [17].

The interoperability consists of technical interoperability, syntactic interoperability, and semantic interoperability. The latter refers to the exchange of data between different IoT applications in a meaningful way. Handling data heterogeneity from different sources and domains is necessary to achieve semantic interoperability. A solution for this problem is the integration of the Semantic Web, which is a comprehensive method for encoding data using a standardized set of metadata [18]. Semantic Web provides a common framework for exchanging and interpreting data between different IoT devices and reduces the risk of data loss. The integration of the semantic web with IoT creates a semantic web of things. The utilization of web semantics is based on the use of some technologies, especially the ontology backbone. Ontologies provide a standard formal representation of IoT data, which facilitates the understanding and interpretation of data between different systems. However, the multitude of specific ontologies defined in this domain creates a significant difficulty in terms of sharing and alignment of knowledge. Most of these ontologies also have been limited to dealing with the data that come from different domains, which is different from the reality where the domains are integrated. In addition to heterogeneous data, vagueness and imprecise of IoT data is another issue. The vagueness and imprecise data in the IoT domain can have significant consequences. For example, in healthcare, it can lead to incorrect diagnoses, ineffective treatments, and negative consequences, such as medication errors or incorrect dosages.

The Semantic Web framework is also based on other semantic web technologies such as RDF, RDFS, OWL, SPARQL, and SWRL rules to enable semantic interoperability. RDF is a standard format for describing resources and the relationships between them. RDF enables the modeling data of sensors and other connected devices. It makes the IoT domain more efficient integration and exchange of data, which can contribute to improved performance. The semantic web of things also provides outputs after processing the IoT data in RDF format. The output can be used as Historical data. The latter is very important for the next treatment, it helps to make accurate decisions, detection of anomalies, and save time and effort by avoiding reprocessing the same data for future use. Using the output of the semantic web of things framework as historical data for future processing requires storing this output in a simple RDF. As the number of IoT devices is very high, the data generated by them is big size, which by transforming it into RDF data, increases the size to reach big RDF data. Despite there are storage methods based on the conversion of RDF data into relational database mode, this conversion led to the loss of some relevant data that influenced the results of the healthcare domain. Moreover, the challenge related to Big IoT RDF Data is not only the storage and management of a variety of data, but also the retrieval time of this data and the quality of these information responses. Finally, consistently to the previous problems, the processing, and storage of IoT data collected from distributed devices using a semantic web necessitate an IoT architecture to ensure scalability and enable appropriate processing at the appropriate time through real-time processing.

Research Objectives and Contributions

Integrating IoT across different domains enables the reduction of production costs, enhancement of the overall quality of services provided, and facilitates better decision-making. Improving this integration requires interoperability between different IoT applications and enhanced capabilities for processing, storing and retrieving the resulting data. In order to improve the framework of the semantic web of things by addressing the problems described above, our thesis provides the following proposals:

- We propose an approach based on IoT architecture and using the semantic web to process data from different sources and domains. An approach is applied to the healthcare and public transportation domains, providing continuous monitoring to patients and health services in the transportation. The approach is also based on an ontology representing both domains and developed using the Neon methodology that provides a quality ontology supporting data sharing and alignment. An ontology that is part of the SAREF ontology is used to represent IoT devices.

- We propose an improvement of the semantic web things framework by enabling the processing of vague and imprecise data. This improvement is achieved by integrating the semantic web with fuzzy logic, which is well adapted to the processing of this type of data, by defining a fuzzy ontology-based approach for representing and sharing IoT data. As the ontology development method is crucial for defining an ontology capable of representing this type of data effectively, the IKarus methodology is used to transform a classic ontology into a fuzzy ontology. The classical ontology is developed by reusing a standard IoT ontology to share and reuse IoT data.This approach is applied to monitor patients with COVID-19 due to their need for accurate diagnosis to facilitate their treatment.

- We propose an extension to the Semantic Web framework to support the storage of their massive RDF output and to retrieve this data quickly. This extension includes classification and indexing layers integrated into the IoT architecture, enabling real-time processing and facilitating the storage of massive RDF data. In the classification layer, fuzzy logic is used to improve data classification, while the binary tree in the indexing layer is employed to facilitate information storage and retrieval. The proposal is applied to monitor patients while the storage structure is associated with their health status.

In all three contributions, we based on the IoT architecture to ensure real-time processing and scalability. In addition, we focused on the reuse of the SAREF ontology to get advantages of its quality and features, such as support extension, alignment, and sharing.

Thesis Roadmap

To facilitate the reading of this thesis, we have structured it into two main parts. The first part is divided into six chapters dedicated to the state of the art and related works. The second part describes the design of the proposals we make, where we have presented three approaches, including the design of the ontologies used for the knowledge representation. The second part also is devoted to implementing and evaluating the proposed approaches. We give in the following a detailed description of these parts:

Part I: State of the Art & Related Work

This part consists of six chapters whose themes are given below:

The first chapter gives an overview of the health domain, beginning with the definitions of health and its domain, the integration of health with other domains as well as digital health and health monitoring, then explaining the vital signals and their importance for the diagnosis and determination of the health status of the patient. In addition, we defined the scoring system and its instances MEWS, and NEWS2. At the end of the chapter, we present the major challenges in the healthcare domain and its need for new technology.

The second chapter presents the state of the art on the internet of things, starting with an introduction, and then we give the definition of the Internet of Things as well as the architecture of the Internet of Things and its components. In addition, it presents the application domain of the internet of things and new paradigms of technology such as cloud, Fog, and Edge computing. This chapter terminates with a presentation of the internet of things challenges, particularly the interoperability challenge.

The third chapter gives an overview of the concepts and principles of web semantics and semantic technologies such as RDF, RDFS, OWL, SPARQL, and SWRL. It presents the backbone of web semantics, the ontology, its origin, definition, and its role, and makes them up, such as classes and relationships. It also presents the development steps of ontology and is terminated by explaining the semantic web of things.

The fourth chapter presents the state of the art on fuzzy logic, its definition and principle, as well as its component. It also presents fuzzy ontology and its development methodology, and the chapter ends with the validation and evaluation tools of this type of ontology.

The fifth chapter of the state-of-the-art part defines Big data, details its characteristic, and explains BIG RDF data. It also presents examples of RDF data management systems. The chapter also presents the systems that implement the different tools, approaches, and technologies presented in this thesis. The chapter ends with explaining the indexing, especially the tree structure.

The sixth chapter presents the works that relate to our approach. It is divided into three parts; the first one presents the web semantics in the IoT domain to solve the homogeneity problem, especially by developing ontologies. It also explains the semantic web of things frameworks that were developed with comparisons and synthesis. The second part presents the works that are based on fuzzy ontology to address the problem of vague and imprecise data. The third part explains the solutions that exist to deal with the storage of RDF data issues.

Part II: Proposed Approach and Results Validation

Part II includes two chapters as below:

Chapter 7: The proposed conceptual model of the semantic web of things framework.

The chapter discusses our three contributions that allow the semantic web of things to solve the problem of semantic interoperability and heterogeneity of IoT data. The three contribution address to health care domain.

The first addresses sharing and alignment problems in the framework of the semantic web of things. It also took into account sharing between domains, such as healthcare and public transportation.

The second contribution allows the improving processing and making of decisions provided by the semantic web of things by proposing a fuzzy ontology. The framework will be able to handle imprecise and vague data.

The last contribution presents an improvement of the semantic web framework by adding classification and indexing layers to allow the storage of their Big RDF data. The contribution enables the storage and retrieval of the outputs of the semantic web of things in a short time. The three contributions based on the proposed IoT architecture include edge, fog, and cloud paradigms.

Chapter 8: Implementation and Experimentation

In this chapter, three applications are showcased, which were utilized to test and authenticate the theoretical approaches discussed in the previous chapter. The chapter provides an outline of the various technologies employed to develop these applications. Moreover, it portrays real-world scenarios in which the outcomes of these applications enable the delivery of health services to users. The chapter also elaborates on the experiments conducted in partnership with medical experts, specifically concerning the first and second applications, to validate the efficacy of the proposed methods. In conclusion, this thesis summarizes the overall contributions and emphasizes the perspectives of this research.

Part I

State of the Art

Chapter 1

Information Driven Healthcare: Leveraging Data to Protect and Improve People's Health

1.1 Introduction

People always need good health and physical comfort to enjoy life. Health allows them to live longer and do all daily activities without the need for the help of others. Because of its great role and influence on the way that people live, it has become very important. People want to protect their health from diseases and have treatment when an illness comes up. For a very long time, people, to benefit from good health and a better quality of life, have used their little experience to transform all available resources to provide diagnoses, produce medicines, detect diseases, etc. They have used natural resources such as plants and animals to invent and produce drugs. They also developed diagnostic tools and built nursing homes.

With the growth of the population and the variety of diseases, people moved to create hospitals and private clinics. They are based on industrial development through the use of the mechanism of the industrial revolution to provide medicines and diagnostic equipment. In addition, they moved to medical education by providing medical schools and universities for research and the training of doctors and nurses. However, the diversity and spread of diseases, viruses, accidents, etc. make these traditional means insufficient to respond to the growing demand to get good healthcare.

In order to face these exigencies, countries provide significant importance to the healthcare domain and make strategies based on providing the domain with adequate human and material resources and encouraging research. To concretizes strategies to solve healthcare problems and provide universal access to healthcare, researchers' attention has recently been on the application of technology.

1.2 Health care

1.2.1 Definition

Healthcare is a domain that aims to ensure the protection and treatment of people, so the health concept should be understood in the first step. The World Health Organization defines health as: *"complete physical, psychological and social well-being, and not merely the absence of disease or infirmity."* [19]. Through these definitions, the health can be divided into three types of health:

1.2.1.1 Physical health

As its name indicates, it is related to the physical aspect, and to achieve it, it is necessary not to eat anything that harms the body but also to avoid behaviors that cause weakness and disease.

1.2.1.2 Mental health

It is related to a person's mental and intellectual aspect and must be preserved by avoiding anything that leads to its destruction or disruption, such as alcohol, drugs, etc. Good mental health is not something the person can have, but something can do. A person with good mental health survives with life problems and contributes to building society.

1.2.1.3 Psychological health

Psychological health is related to the spiritual aspect of a person. Doctors recommend avoiding stress, isolation, fears, sickness, and other psychological problems to make humans feel good and happy. Neither mental nor physical health can exist in isolation (a healthy mind resides in a healthy body, and vice versa), but physical and psychological functions must depend on each other.

Health care is all health services delivered to people to preserve the health of healthy people and restore the health of sick people through treatment and rehabilitation. It refers to all material and human resources as well as health information destined to protect the healthy person and provide patients with health services [20] [21]. Health care is not just concerned with protecting human health and ensuring the availability of basic goods and services in the case of illness, but also with improving the quality of life and rendering it more comforting and happy [22].

1.2.2 Integrating Healthcare with Other Domains to Improve Outcomes

Since health is very related to the individual and the population, it is associated with all their daily activities in different domains such as education, transportation, work, tourism, sports, etc. These activities are capable of influencing human health in a positive or negative way. For example, sports can improve a person's health. Other domains are also impacted by the state of human health. For example, health conditions during the COVID-19 period had a negative impact on the global economy [23]. Since any improvement in health is directly due to the development of various other domains. They began to give great importance to a person's health, whether a worker or a consumer, by providing health services and a healthy environment, etc. For example, workers in institutions or factories benefit from ways to protect their health and provide assistance in the case of illness, such as getting reduced prices for drugs.

To improve and develop other domains; it is necessary to improve the impact of health care in other domains; this improvement is started by using computer science capabilities.

1.2.3 Digital Health

The term "digital healthy" refers to a set of emerging technologies utilized in the fields of health care, health information, and other terms like m-health and e-health. In 1879, telephone calls were seen as an important way to facilitate medical consultations for patients. In 1925, science and invention reported on the then-rare use of radios to educate the broader population about health issues [24]. Since computer science invention and using it in health care, It enabled to generate the health care informatics. Computer science provides a system and techniques that enable the acquisition, treatment, exchange, storage, and retrieval of health information. Computer science starts to use in healthcare with administration tasks to develop other tasks. Computer science benefits from tools like scanners, cameras, etc., to acquire data and also from other tools like print, screens, and others to show its output. It creates a change in the healthcare domain and enables optimizing time and facilitating tasks. The healthcare domain also benefits from the internet and related technologies to provide a new way of working [25]. It becomes getting health information and providing its services remotely. As depicted in Figure 1.1, it allows distant medical consultation, remote assistance, and remote patient monitoring that gain the time, decrease overlapping in hospitals and decrease the costs.

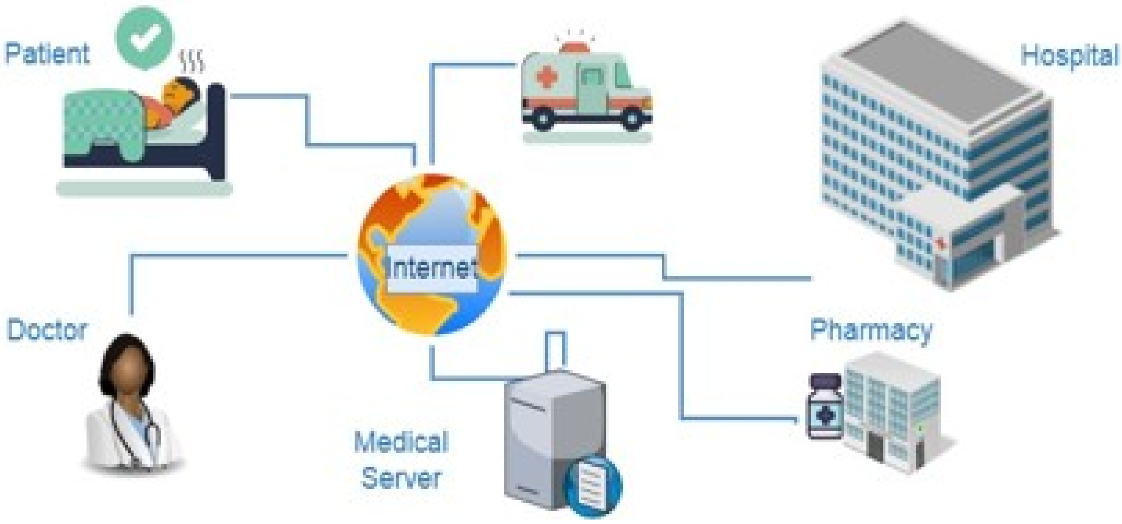

Figure 1.1: Remote healthcare.

1.2.4 Health monitoring

In the healthcare domain, many people need to be examined almost daily. They are elderly people or children as well as suffering from a disease as a chronic disease. These people are more at risk than others of suffering complications or deterioration in their health that may lead to death. On the other hand, health care cannot achieve its intended purpose without handling and addressing the needs of these particular people. Health monitoring is an essential part of the healthcare field aimed at providing special health services represented by continuous monitoring of the health status of the patients. It is based on the observation of health conditions that may affect a population and enables the detection of any change associated with the health of the patient to protect them from deterioration of their health status [26].

As the number of these patients grows, the effort of using nurses or assistance to monitor them becomes impractical and insufficient. Defining systems and developing software services that collect, exchange and evaluate health data has become a necessity for effective health monitoring. It can help the healthcare sector by using doctor knowledge to facilitate treatment, speed up medical intervention and reduce costs and time. In health monitoring, doctors based on the measurement of data from the body of patients to take decisions about their health status. These data are easily collected from outside of the human body and do not demand expensive equipment.

1.3 Vital signs

Vital signs are a set of measurements of the essential physiological functions of a living organism. These measurements are essential for identifying or analyzing a person's health problems and are the first step in any medical consultation. It can be measured in a doctor's office, hospital, emergency room, home, or elsewhere. The normal ranges for an individual's vital signs are dependent on factors such as age, weight, gender, and overall health status [27]. In the following, we will find a statement of the four main vital signs that are usually measured periodically and regularly.

1.3.1 Body temperature

This temperature is an important indicator of a person's health status. To evaluate a patient's health status and to understand the relationship between temperature and disease, it is important to define a baseline body temperature, the "normal temperature". Healthy adults' average normal body temperature varies between 36 and 38 Celsius [28]. It can vary depending on several factors, such as the amount of food and fluids consumed, physical activity, gender, etc. It also varies depending on the area of the body being measured; for example, the body temperature measured from the area under the armpit is slightly lower than that taken from the mouth.

1.3.2 Heart rate

It is a crucial sign that indicates a person's condition of health. It describes the number of heartbeats, beats per minute, or rate of heartbeat. While the average adult heart rate is between 60 and 100 beats per minute, certain persons may have a lower or higher resting heart rate. Normal heart rate is influenced by factors including body posture, such as standing or sitting, amount of activity, age, weight, and medications used.

1.3.3 Respiration rate

It represents the number of breaths taken in one minute. It counted by observing the swelling of the chest in minutes or by using a respiratory rate monitor. Adults' normal resting respiration rates normally fall between 12 and 20 breaths per minute and change when patients have a fever, exercise, and have certain diseases. A fast or slow respiratory rate can also sometimes indicate a serious respiratory problem. The respiratory rate also helps to detect other important indicators that can be measured in addition to respiratory rates, such as depth of breathing and pattern of respiratory rate.

Score	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Systolic BP		>199		100-199	80-99	70-79	<70
Heart rate	>129	110-129	100-109	50-99	40-49	30-39	<30
Temperature		>38.9	38-38.9	36-37.9	35-35.9	34-34.9	<34
Respiratory rate	>35	31-35	21-30	9-20			<7
Oxygen saturation	0-84	85-89	90-94	95-100			

Table 1.1: Modified Early Warning Score [15]

1.3.4 Blood pressure

It is an essential vital indicator that can be used to assess a patient's blood circulation. It is defined as the force or pressure imposed by blood on the walls of arteries as it flows through them. Blood pressure is measured in millimetres of mercury (mm Hg); the average normal blood pressure in young, healthy adults is between 90/60 mmHg and 120/80 mmHg [29]. These vital signs can be used by an early warning system to quickly determine the degree of illness of a patient.

1.4 Early warning score systems

Early warning score systems are a quick way to find persons who are deteriorating and who have abnormal physiological traits. Early warning scoring systems are designed to prevent people from passing away or deteriorating into grave health conditions. The most popular approaches for determining the health status of individuals, particularly those in hospitals, are as follows:

1.4.1 Modified Early Warning Score (MEWS)

The Modified Early Warning Score (MEWS) for clinical deterioration is a scoring system for physiological parameters, including heart rate, blood pressure, body temperature, and respiration rate [15]. It was created by the Royal College of Physicians to give British hospitals a uniform national standard. The scoring system has a score between "0" and "3", and the patients with a total MEWS score ≥ 5 or a score equal 3 for any of the physiological parameters require a higher and faster level of care(see Table 1.1).

The Modified Early Warning Score can be used to categorize patients into different risk groups according to their present health state (EWS). The classes associated with the Modified Early Warning Score are illustrated below, along with recommended actions for each class:

- 0-2: Low risk, monitor frequently.
- 3-4: Moderate risk; notify the medical team; consider increasing the level of care.
- 5-6: High risk, medical emergency, notify medical team, transfer to a higher level of care.
- 7 or more:Very high risk, medical emergency, and notify medical emergency team.

1.4.2 The National Early Warning Score

The National Early Warning Score(NEWS) is a verified tool created in the United Kingdom (UK). It detects early indicators of deterioration and initiates the appropriate responses using seven physiological markers that nurses currently regularly monitor when providing patient care. The National Early Warning Score comprises 7 parameters (heart rate, systolic blood pressure, temperature, saturation, respiratory rate, state of consciousness, and presence or absence of Oxygen). A score of 0 to 3 is assigned to each of the seven parameters, then the evaluation result is calculated [16]. The probability of worsening increases with the NEWS score arises. In a hospital, Nurses or doctors are urged to keep an eye on the results and take data from other assessments into consideration when making decisions rather than concentrating on a single NEWS score to ensure that each patient's "normal" values are considered.

Table 1.2 present an extension of NEWS is called NEWS2 [30]. The National Early Warning Score two makes many changes to the NEWS vital sign weights and includes a new SpO2 two rating scale that is used in patients with confirmed type II respiratory failure. A comparison between the two existing scoring systems existing in [31]. For patients, within 24 hours, NEWS and MEWS are more reliable at predicting in-hospital cardiac arrest, death, and transfer to intensive care units. Both systems only need timely information, enabling them to make decisions and act appropriately. In order to reduce the time, expense, and effort involved and to improve the experience, fundamental healthcare challenges must be identified.

Score	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Systolic BP	<90	91-100	101-110	111-219			<220
Heart rate	<40		41-51	51-90	91-110	111-131	>131
Temperature	<35		35-36	36-38	38-39	>39	
Respiratory rate	<8		9-11	12-20		21-24	>25
Oxygen saturation	<91	92-93	94-95	>96			

Table 1.2: National Early Warning Score(NEWS)2 [16]

1.5 Healthcare Challenges

Many challenges decrease the healthcare efficiency to prevent people from disease and provide those sick with adequate health services. They put a life of many peoples at risk and require providing more effort and cost to achieve the satisfaction of the patients. In the following, we detail the fundamental challenges of the healthcare domain:

- *The world population has grown rapidly*
  The world knows a highly growing number of people. This growth requires high resources in terms of personnel such as doctors, nurses, and administrators, etc. In terms of material resources, such as the number of hospitals, ambulances, medical equipment, and others. Moreover, this number of patients needs a high quantity of drugs for treatment. The increased use of these resources results from the field's rising prices, which pose a problem for governments, particularly those with little income [32].

- *The high number of diseases and the spread of the virus*
  The healthcare domain knows the number and multiplicity of diseases. Some of them require a long time for treatment, and others need more resources. These maladies increase the number of patients that consume all healthcare resources. In addition to disease, the quick spread of some viruses create great problem in the health sector, where the virus caused the death of hundreds of doctors and nurses [33].

- *The rising number of persons who has a chronic disease*
  Chronic diseases are caused by physiological, genetic, environmental, and behavioral factors. Chronic disease is characterized by a long period of illness, which means that long-term treatment is required. In the USA, six from ten adults have a chronic

disease, and four in ten have two chronic diseases [34]. Patients with chronic diseases take up most of the healthcare costs and need continuous health monitoring.

- *Limitation of financial resources to encourage research and innovation*
  The financial resources of health care in some countries are more than in others, especially in countries with good economies. Health care requires more financial resources to provide adequate health services. For example, the USA spends about $4.3 trillion or $12,914 per person [35]. Furthermore, these financial resources contribute to innovation and research in health care.

  In addition to these challenges, other factors lead to spreading disease and increasing health care costs like poverty, pollution, weather changes, etc. To overcome these challenges, we need a low-cost solution appropriate to the number of patients and the economic state. A solution that is adapted to the specification of the domain in such a way that the treatment will be easy and available for all patients, as well as respect the time factor that relates to the patients' lives. The solution should be effective in a way that takes advantage of current technologies and improves traditional treatment methods.

1.6 Conclusion

This chapter provides an explanation and details about the healthcare domain, particularly the definition of health, digital health, and the health monitoring of patients. The chapter also presents the vital signs as important collected data from patients and can be used for diagnosis to detect their health status. As the use of vital signs needs the system to read and facilitate the interpretation of its value correctly by defining the range's value of each vital sign, the two early warning score systems MEWS and NEWS2 are explained. Finally, the chapter defines a set of challenges that prevent the healthcare domain from providing quality health services. The fundamental challenges are explained and occurred the need for solutions ensures the continuous monitoring of patients with low-cost tools. A solution consists of adopting the advances in science and technology to address the multidisciplinary tasks of healthcare, and to ensure that the healthcare domain contributes to the advancement of the other domains. One of these technologies is the internet of things, which can provide remote patient monitoring, increase efficiency, etc.

Chapter 2

The IoT Evolution

2.1 Introduction

Today, technology plays a crucial role in improving the quality of life of people in the world. Most of these technologies are based on the internet, which facilitates communication between different machines. Despite its invention in 1969 by ARPA – Advanced Research Projects Agency, it has grown with the birthing of the Web that occurred in the 1990s to ease using and exchange the data [36]. It has become an essential factor in advance of any domain, economic, social, education, etc. It is due to the high development of communication and technologies. One of this technology that makes a real revolution in the world is called the "internet of things". The internet of things enables any physical thing to connect to the internet, collect data for its surroundings, and then process and exchange it with other devices. The internet of things impacts different domains, bringing them more capabilities and vast opportunities such as cost and time. This growing interest in adoption in different domains and the increasing number of people using connected devices requires the emergence of new technology that enables the storing, computing, and analysis of its data providing adequate communication.

Cloud Computing, fog computing, and edge computing are new emerging data storage and processing platforms. The technology Cloud computing is essential for the IoT evolution because it provides unlimited storage and processing power. However, its limits are related to the latency time and the high cost due to the invention of other technologies. Fog computing is a cloud computing extension that helps bring various services closer to users. Edge computing is another paradigm developed to ensure real-time, especially in such a domain where time is more important, like health care. In this chapter, we will provide an overview of fundamental IoT concepts. We will cite their definitions, architecture, and application domains, especially the healthcare domain. Then we will illustrate the evolutionary paradigms of cloud computing, fog computing, and edge computing. At the end of this chapter, we will explain the main challenges, especially the interoperability challenge.

2.2 Internet of things

2.2.1 Definition

The Internet of Things (IoT) is a new concept that has been coined and developed in recent years. It was proposed by Kevin Ashton, executive director of the MIT Auto-ID laboratories the first time in 1999 [37]. He started by attaching RFID tags to objects to allow connection and data exchange and becomes believed the internet of things has the potential to change the world. *"The internet of things has the potential to change the world, just as the internet did. Maybe even more so."*

As the Internet of things has missed a standard definition in the literature, we cite some of the most frequent definitions: IEEE definition [38] explains the term "things" as an abject with a unique identification that enables sensing and actuation with the surrounding. It confirms that identification has the crucial rule for allowing data exchange:

"An IoT is a network that connects uniquely identifiable 'Things' to the internet. The 'Things' have sensing/actuation and potential programmability capabilities. Through the exploitation of unique identification and sensing, information about the 'Thing' can be collected and the state of the 'Thing' can be changed from anywhere, anytime, by anything."

Atzori et al. [39] defines IoT as a result of the evolution of wireless telecommunication technology. It allows various things around us to interact with each other by using a unique address.

"The internet of things is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications. The basic idea of this concept is the pervasive presence around us of a variety of things or objects - such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc. - which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals."

According to oracle, the internet of things is represented by all objects that can connect and interact with others. The objects are embedded with sensors, software, and other technologies to enable the communication through internet: *The internet of things describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household ob-*

jects to sophisticated industrial tools. [40]

Recent definitions of The internet Engineering Task Force (IETF) have defined IoT, and the definition is based on the ability of the IoT technology to exchange the data with different actors such as manufactory, operator, and/or other connected devices as [41] : *The internet of things is the network of physical objects or "things" embedded with electronics, software, sensors, actuators, and connectivity to enable objects to exchange data with the manufacturer, operator, and/or other connected devices.*

Based on the above definitions and others, we can define the internet of things as *"An emerging paradigm that equips things with electronics, software, and sensors to transform them into intelligent objects capable of collecting and exchanging data anytime, anywhere, and with anyone via the internet to ease of decision-making"*.

2.2.2 Internet of things functioning

To operate, internet of things needs to go through four phases, as stated by Perry Xiao [42]: Due to the evolution of Internet Protocol (IP) addresses and their generations, the internet gives hundreds of different IP addresses. The first phase entails assigning a unique identifier to each "thing" that will be connected to the internet of things. The "things" must connect with one another in the second phase, using existing communication technologies such as WiFi, LoRaWAN, Bluetooth, ZigBee, and others. The third phase consists of equipping the things with a sensor appropriate for their environment and the type of measurements for providing data collection. The fourth step comprises data processing, sensor management, and communication via devices such as microcontrollers.

2.2.3 IoT architecture

Many IoT architectures are defined in the literature, and most of them consist of three [43] or extend to five layers [44]. The Perception (or sensing) layer, the Network layer, and the Application (or Transmission) layer are the three commonly defined layers. The processing and business layers are added to the previous architecture to provide more efficiency, adequate for the rapid growth of the internet of things. Each of these layers that are present in Figure 2.1 is defined within its functions and devices.

2.2.3.1 Perception Layer

The perception layer is also known as the "Sensors layer " in IoT. The layer consists of devices like sensors, RFID tags, actuators, cameras, Global Positioning System (GPS), laser scanners, and wearable devices. It allows the detection and collection of the data

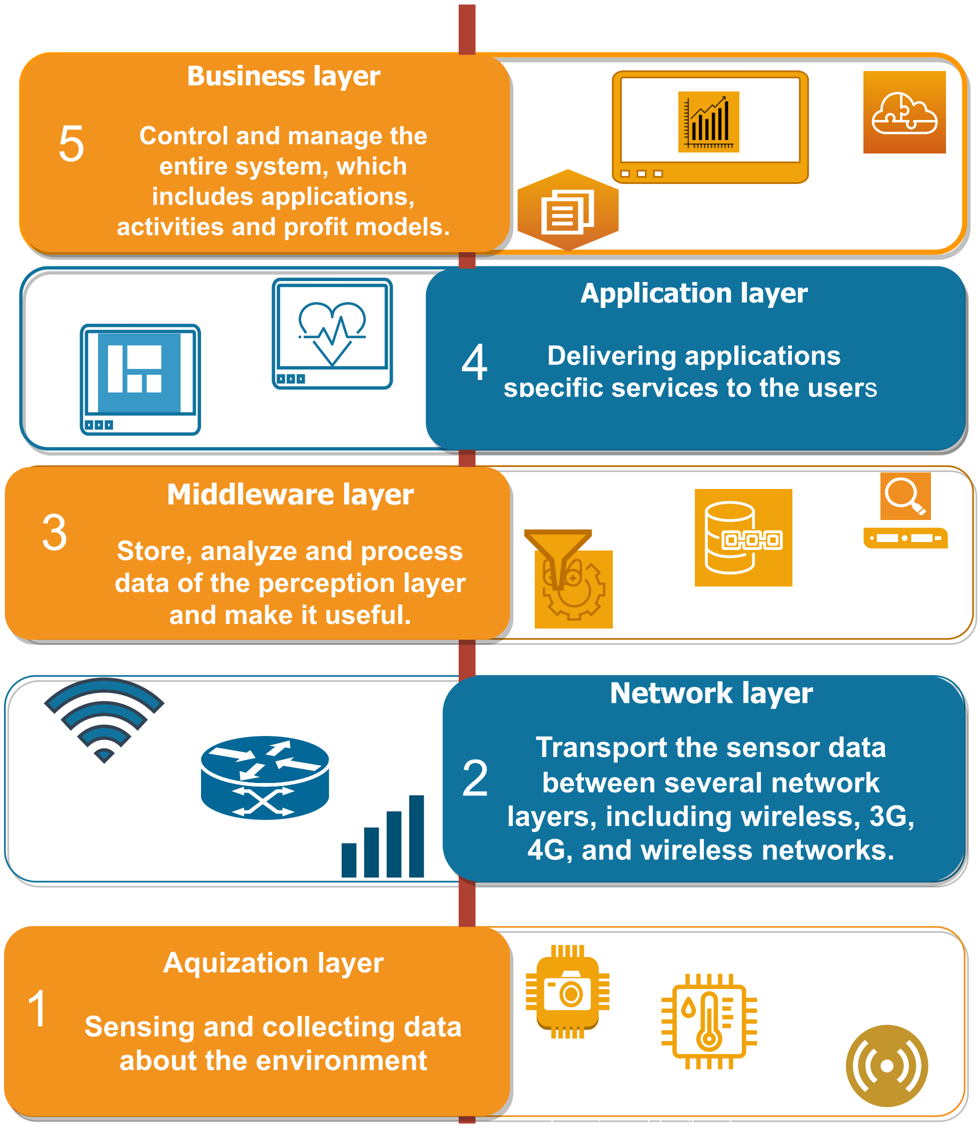

Figure 2.1: Five-layer architecture of IoT technology.

from the environment (such as light, temperature, humidity, human body vital signs, etc.) and transmit it to the network layer. It also supports collaborations with other IoT devices in local and short-range networks.

2.2.3.2 Network Layer

It is the next layer after the perception layer responsible for the interchange and sharing of information are collected. It is also called the "Transmission Layer" and "transport layer". To ensure safe transmitting, it consists of several network devices, smart objects, and servers allowing connection to the internet or locally using recent technologies such as WiFi, LTE, Bluetooth, 3G, Zigbee, etc.

2.2.3.3 Middleware Layer

It is also known as the "processing layer." That is responsible for processing, storing, and analyzing the perception layer data and making it useful. This layer supports managing lower layers and links to the database. It can benefit and employs different technologies such as cloud computing. The layer also can provide decision-making based on its calculation of perception data.

2.2.3.4 Application layer

It is the layer responsible for interpreting the processing layer to customer services delivered to users. The layer provides an application service appropriate to deployments such as smart homes, smart cities, smart agriculture, smart transportation, etc.

2.2.3.5 Business Layer

The business layer is at the top of the IoT architecture, responsible for creating flowcharts and graphs, analyzing results, defining the business model, and ensuring user privacy and confidentiality, which enables controlling and managing the entire IoT system.

2.2.4 IoT Communication Technology

Communication technologies were defined to allow objects to send and receive data over a network. In the IoT domain, many technologies can be used for IoT communications, that depicted in Figure 2.2, such as Ethernet, Wi-Fi, ZigBee, and Bluetooth. Each of these communication technologies has its characteristics, such as manner of communication, distance of communication, speed, cost, etc, which makes it suitable for the implementation and the environment in IoT applications [45]. The syntax, synchronization of communication, and error retrieval methods are associated with a set of rules called communication protocols. The protocols allow the devices to connect and exchange data with others. A close analogy to protocols is that of human languages. For IoT applications, there are many communication protocols available. The Commonly used protocols include CoAP, Websocket, XMPP, and MQTT.

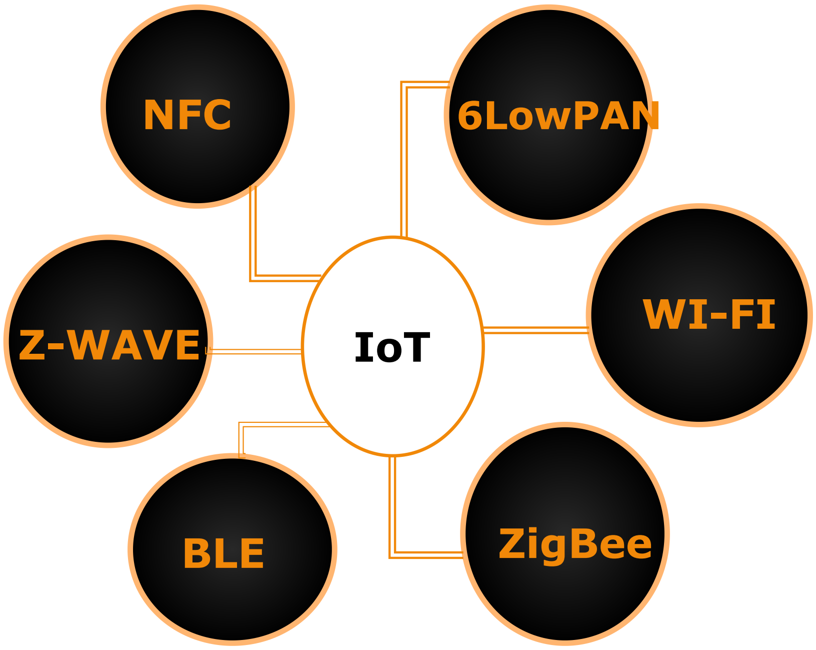

Figure 2.2: Example of communications technologies of IoT.

2.2.5 IoT platforms

IoT platforms facilitate the integration of the IoT device with the IoT application to make physical objects online and provide the users with its services. IoT platforms allow IoT tasks like collecting, visualizing, and analyzing. They ease device management in addition to offering the security and authentication of users. In the IoT domain, several IoT Platforms are available, including over 300 platforms [46] [47]. The most popular that provided by the big cloud such as Amazon Web Services (AWS) IoT platform by Amazon, Google Cloud Platform IoT by Google, IBM Watson IoT platform by IBM, Microsoft Azure IoT platform, and other open source solutions like OpenIoT.

2.2.6 Internet of things applications

To enhance people's daily lives and activities, IoT can be used in a range of domains, such as the home, health care, industry, transportation, and surveillance.

2.2.6.1 Smart home

The smart house, also known as home automation, is an extension of building automation. It is based on wireless sensor networks and actuation technologies at home, making it safer and more comfortable and providing a high quality of life. As well as IoT, many things in the home have become smart, such as light control, temperature management, surveillance camera, remote control, etc. It connects equipment to automate numerous daily tasks, such as turning on and off heating systems and starting and stopping cooking and washing. It ensures its autonomy surveillance by detecting the intrusion and acting its alert systems [48]. The smart home provides its user with different information, enabling them to remote control through smartphones or laptops. The smart home also eases the monitoring of older people, patients, and family members remotely and provides them with adequate services [49].

2.2.6.2 Smart transportation

Transportation is concerned with organizing and physically moving items, people, and services to make their performance easier. The internet of things has the potential to improve transportation infrastructure significantly. It provides accurate data that helps drivers make smart judgments while travelling, find parking, and avoid traffic jams. It also encourages several firms, such as Tesla, Google, Volvo, Volkswagen, and General Motors, to develop electric vehicles. The internet of things has changed vehicles to become smart, connect with others, avoid accidents, and be more autonomous. Furthermore, the internet of things improves logistical activity by enhancing visibility from beginning to end, warehouse management, and monitoring quality conditions throughout goods shipment. Smart transportation includes smart public transportation [50] that assists passengers in planning their trips by providing regular updates and safer transportation. It allows agencies to increase profits by providing timely services and comfortable travel and satisfies travellers' desires.

2.2.6.3 Smart Agriculture

Agriculture is the world's primary source of food. At the same time, it is the world's largest water consumer, suffers from diseases, requires more workers, and is monitored continuously. The most appropriate way to solve these issues is to integrate IoT technology. It gives farmers the tools to make better decisions and automate processes by delivering goods, knowledge, and services that boost production, quality, and profit. The internet of smart farming is a real real-world example of the integration of the internet of things. It enables management and controls the farming remotely. The internet of things can track rain, ice formation, drought, and other weather conditions that affect farming [51].

2.2.6.4 Smart industrial

The smart industry, also known as Industrial IoT (IoT), is built on integrating machines, software, sensors, and the internet to enable automation and offer factory intelligence, allowing machines to self-configure and make decisions with little human intervention. Smart industries enable faster detection and resolution of issues, saving money and time [52].

2.2.6.5 Smart environment

Using IoT by deploying environmental sensors allows protection from natural disasters such as earthquakes, landslides, etc. The IoT provides the detection, transformation, and processing of information that ensure quick intervention and taking precautions through warning systems. The internet of things can be used to monitor environmental conditions, such as detecting CO2 emissions from factories and cars, monitoring the quality of air, water, and soil, detecting insects in forests, etc [53].

2.2.6.6 Smart healthcare

Healthcare is an essential domain related to people's lives. It is about protecting them from disease and ensuring wellness through an infrastructure that enables them to provide health services. Smart healthcare, also known as "health IoT" or "Internet of Health Things" is adopting internet of things technology in healthcare domain infrastructure to make it more intelligent. The internet of things was adopted for medical devices to provide them with new capabilities to improve their performance [54]. It enables the collection of various data about the patient's health, clinic management, health activities, etc(see Figure 2.3). It provides the capabilities to store these data, transfer it with other machines, or share it with medical actors such as doctors and administration. The internet of things also enables the processing of the data collected, helps to make decisions, and provides health services. Using the internet of things in health clinics decreases human administration costs and helps improve treatment results through remote patient monitoring.

Smart healthcare systems allow patients remote monitoring of health status without moving to the hospital, reducing the cost, effort, and time [48]. It provides them with early diagnosis and ensures good diagnosis by exchanging their data with doctors, hospitals, and laboratories to get correct information and results. It offers them more control over their lives and treatment at all times. Healthcare systems can also deal with the rapidly growing number of elderly people around the world. It provides them with remote health monitoring and quick intervention through emergency notification systems when they are in serious health conditions. The physicians in smart health care become more approximate to patients. They can make the correct diagnoses and make efficient decisions at appropriate times. They do not need to move to hospitals for each work and can do it remotely; furthermore, they can share information with hospitals, health organizations, etc.

Smart healthcare changes the traditional clinics and tools using various IoT mechanisms and advanced data analysis techniques to create new smart health things such as intelligent hospitals, smart pharmacies, smart ambulances, etc. These health clinics limit human intervention and overcome the overloading of patients by reducing the number of them needed to visit them, particularly during the pandemic periods. Patients with chronic diseases need to visit health clinics monthly or daily to monitor their health status. They benefit from health services based on measuring their vital signs such as blood pressure, heart rate, body temperature, blood sugar, etc.

Figure 2.3: Smart healthcare system.

2.3 Cloud computing

Google and Amazon coined the term "cloud computing" in 2006 to describe a collection of high-performance servers that provide on-demand services to internet customers. Cloud Computing refers to the system hardware and software in the data centers provided over the internet to offer services. The National Institute of Standards and Technology (NIST) published a widely used and referenced definition of cloud computing in 2009 [55]. "Cloud computing" is defined as *"a set of disciplines, technologies, and business models for delivering computing capabilities (software, platforms, and hardware) as a service on-demand"*. Cloud computing provides unlimited storage space, and power processing enables data pre-processing, data analysis, decision-making, and a variety of other services adapted to the request of users. Most cloud computing is a dominant business in that its services are paid, and its clients can access it from any device connected to the internet in any place. Service-Based IT-Resources, On-demand self-service, ubiquitous access, multitenancy, location independence, and rapid elasticity are all features of cloud computing.

Cloud computing supports the advancement of the internet of things and improves the quality of services provided by its application. It provides the necessary tools, hardware, or software to process the massive amounts of data generated by IoT devices and enables the exchange of its data in a large zone (see Figure 2.4).

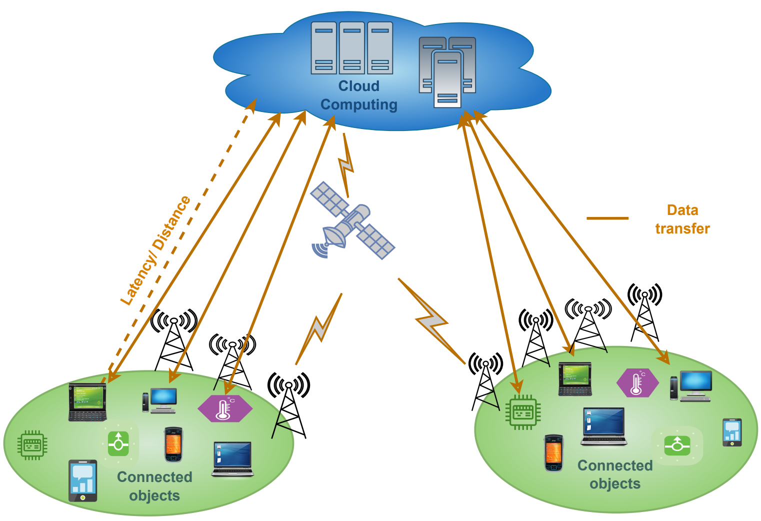

Figure 2.4: Cloud computing approach [1]

The main cloud computing services are divided into three categories [56]:
The first category is called SaaS (Software as a Service), which allows cloud customers to

use software without having to worry about the underlying stack (application platform, hardware, etc.) or the software installation, such as an email service. This category allows the customer who creates and uses the data to share data control with the cloud provider who hosts, stores, secures, and backs up the data.

The second category is called the PaaS (Platform as a Service). It allows developers to build and manage applications without having the need to deal with the infrastructure (application platform, operating system, hardware, and network).

IaaS (Infrastructure as a Service) is the third category. An IaaS is a solution that provides users with lead computer infrastructures, such as virtualization, servers, networks, and data storage. This Cloud Computing service necessitates administrative skills and autonomy to manage the operating system, applications, and data, among other things.

2.3.1 Cloud computing models

Depending on their unique demands and goals, enterprises can select from a variety of cloud computing models. These models consist of: [57]:

2.3.1.1 Private cloud

Exclusively makes all resources available to a single customer(enterprise) or a group of consumers. The cloud is controlled by its owner. It provides cloud services with a higher level of security and privacy through using the firewall and internal hosting.

2.3.1.2 Public cloud

It is the most common model of cloud computing. It allows users to access cloud services over the internet without knowing where their information is stored or processed. It also allows a user's computing resources and databases to be hosted in any of the provider's data centers.

2.3.1.3 Hybrid cloud

Combines public and private cloud services. For security, a hybrid cloud offers each cloud to be controlled independently and, at the same time, allows the data and applications to be shared among the clouds.

2.3.2 Cloud advantages

The advantages of this paradigm will be presented as follows [58]:

- Flexibility is one of the key advantages of cloud computing, as it allows for using multiple resources during application execution. This provides the possibility to request additional resources if necessary.
- Cost optimization is achieved through a reduction in computer headcount and pricing models that are based on the duration of computer resource usage without the need for heavy initial investments.
- Portability is a key feature of cloud computing, as it allows organizations to access their computing power from anywhere in the world where users have an internet connection.
- The simplicity of use is one of the benefits of cloud computing, as it offers applications and services that are easily accessible through web pages and do not require complicated installations.

2.3.3 Cloud limitation

Cloud computing creates a revolution in the IoT domain. It provides users with different services, especially those requiring high processing performance and large storage capacity. Cloud computing in the beginning can provide these services at an acceptable cost and short time.

Many companies demand these services, and cloud technology has become essential for their work due to the growing number of users and the mounting of data transferred. Furthermore, some IoT applications, such as smart cities, vehicles, and others, require real-time processing. As a result of this state, some limitations are due to reducing its efficiency. In the following, we present the principal of these limits [59]:

2.3.3.1 Latency

Cloud computing provides its services to users connected via the internet. Most of the devices communicating with it have a long distance. It requires a significant amount of time for a packet to travel from one place to another and back. Most internet of things systems require an adequate response speed. Cloud computing technology cannot guarantee this demand at all times.

2.3.3.2 Unused Cloud Computing Resources

Cloud computing provides users with sources that enable high processing and large data storage. As more data and applications are on the cloud, the greater the potential for waste, as keeping track of them becomes increasingly challenging. This state increases the demand for new resources required for each new cloud application launched or redesigned, which cannot always suit the application's requirements.

2.3.3.3 Bandwidth bottleneck

As the internet of things technology becomes more and more in high demand, the number of devices generating data also grows. The use of the cloud in processing and storing these data creates a significant challenge related to the bandwidth capacity called "bandwidth bottleneck". This challenge does not permit data transportation between the cloud and the users, preventing them from benefiting from its services.

2.3.3.4 Geographic distribution

IoT devices are distributed in different locations. Centralized cloud computing requires transferring all data to the same point for processing or storage. Many IoT applications have many requirements, such as real-time interactions, domain property, etc. However, the cloud infrastructure has difficulty providing it.

2.3.3.5 Security

Once multiple devices generate IoT data, it is transferred with other data to the cloud for processing and storage. The long-distance between these devices and the cloud requires it to be traversed by other devices using a communication channel, so it can be compromised and modified. With the use of cloud technology, it is crucial to ask questions about the security of the information.

2.3.3.6 Internet availability

Users, to benefit from cloud services, must connect via the internet. The internet allows IoT devices to transfer the data they generate and receive the results from the cloud. Any loss of the internet connection leads to the loss of cloud services and the disappearance of the generated IoT data. IoT devices need technology to provide similar cloud services to avoid any consequences of a possible connection interruption. As these challenges influence on the advancements of cloud computing, researchers believe it is necessary to propose new technology enables the extension of cloud computing.

2.4 Fog computing

At CISCO 2012, Flavio Bonomi defined the term 'fog computing' as a computing extension of cloud computing [60]. It proposes overcoming the limitations of cloud computing by using cloud capabilities near IoT devices. Fog computing addresses the needs of applications that require cloud computing services, such as storage and processing, to be closer to customers. It is based on a decentralized computing architecture that allows for geographic distribution, mobility, and maintaining scalability to ensure using of large numbers of devices.

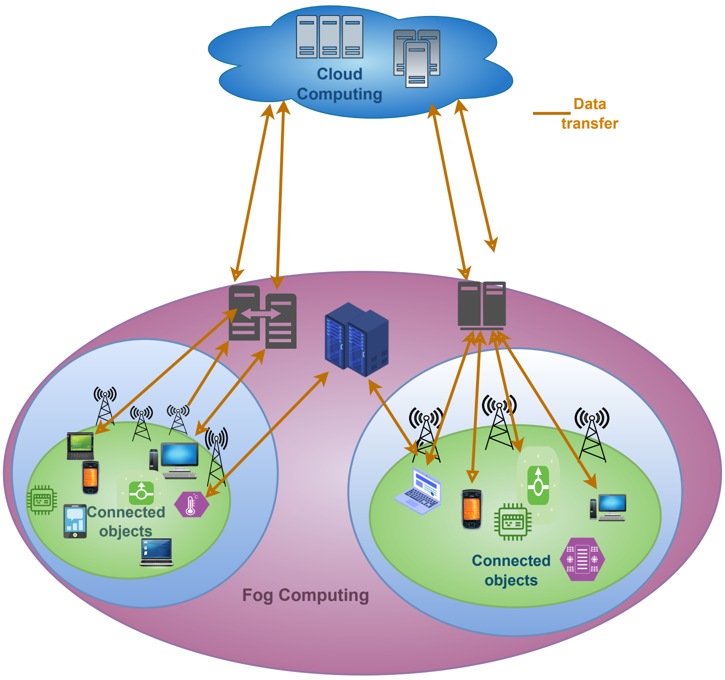

Figure 2.5: Fog Computing approach [1]

Iorga et al. 2018 proposed a detailed definition of fog computing, covering its infrastructure, exploitation, and benefits compared to the cloud [61]. They defined *"fog computing as a layered model for enabling ubiquitous access to a shared continuum of scalable computing resources. The model facilitates deploying distributed, latency-aware applications and services and consists of fog nodes (physical or virtual) residing between smart end devices and centralized (cloud) services. Fog computing reduces latency, decreases bandwidth, and improves dependability by extending computing, storage, communication, and networking functions to the network's edges.*

Through this definition, fog computing represents a harmonization between data centers and end-users to easily manage and program computing, networking, and storage services (see Figure 2.5). Fog computing is also characterized by its facilitating of use through the network. It can be used on a single node or across multiple nodes, and its resources can be integrated into access points, routers, and network gateways alongside standard network activities. Many new IoT applications and services have been suggested using fog computing's characteristics. It is well suited to traffic safety, e-Health, web content delivery, augmented reality, etc [62].

2.4.1 Fog advantages

The birth of fog computing has brought many advantages to the IoT domain. In the following, we will cite some of the main benefits [63]:

2.4.1.1 Low latency

Many IoT applications demand low latency, and using the cloud is inadequate. It has a great distance, with IoT devices requiring time to transmit and a significant amount of unprocessed data needed to transfer to it for treatment. Fog computing prevents this transport action of these big data by offering cloud features, including processing, filtering, and storing the data close to the source. It avoids the undesirable number of hops a package must make before obtaining a response and decreases the data supply to cloud servers. Fog computing guarantees high IoT applications' response that is increased the efficiency of these applications.

2.4.1.2 Mobility

Different to the cloud computing, fog computing is decentralized. It can be installed everywhere to offer dispersed services and applications near end-users.

2.4.1.3 Geo-distribution and location awareness

Fog computing supports mobility when one can move from one location to another without losing the fog services.

2.4.1.4 Scalability

Fog computing is a multilayered distributed system that helps manage the fast expansion of the internet of things. The scalability enables the IoT application to save the quality of service with the increasing number of connected devices, application features, and users.

Other benefits of fog computing include real-time interactions, a preference for wireless access, support for online analytics, and cloud integration.

2.4.2 Fog computing in healthcare

Fog computing enables data processing by IoT devices in real time with minimal overhead. This approach can address latency and rapid response issues, which are critical for healthcare applications. Fog computing is used in many works to help detect the disease and track patients.

Cao et al. [64] propose FAST, a fog-computing assisted distributed analytics system, to track falls in stroke patients. They developed a fog-based real-time fall detection system that divides the duty of fall detection between edge devices and the cloud. Using a data set created in 2010, Singh et al. [65] propose a fog architecture-based methodology to detect and monitor the dengue virus by classifying dengue patients into uninfected, infected, and seriously infected. This framework aims to produce a latency-aware system for categorizing users into various categories based on their symptoms. In order to create an efficient architecture for healthcare and elderly-care applications, Stantchev et al. [66] presented a three-tier architecture for a smart-healthcare infrastructure, consisting of a role model, layered-cloud architecture, and a fog-computing layer. The fog layer provides low latency, mobility assistance, location awareness, and improved architecture. For handling security issues at the fog computing level, Rehman et al. [67] proposed a secure health fog framework that collects data from different internet of things devices. To defend against the attacks, they proposed a protocol that employs symmetric and public-key cryptography.

Despite Fog computing also enabling local storage, filtering, aggregating, and local extracting of essential information to respond to the user request or other systems, the need to bring data processing to end devices is due to the occurrence of other new technology.

2.5 Edge computing

Edge computing is a distributed computing paradigm for IoT environments in which computing resources, storage capacity, and processing power are as close as possible to the end devices and data-generating sensors. This approach provides an alternative to cloud computing solutions with centralized servers(see Table 2.1).

The basic principle of edge computing is to move applications, data, and services from the cloud to the edge of the internet to address the challenges a centralized cloud system pose [68]. As shown in Figure 2.6, edge computing is a fog computing alternative based on the idea that computation should always occur close to a data source. Edge computing focuses on the practical side, while fog computing focuses on the substructure.

Edge computing includes various features, including local data processing, data filtering, data cleaning, data transfer, and local data storage for local use. In this context, Edge computing has gained widespread acceptance as a crucial element of IoT systems, particularly in handling time-sensitive applications and minimizing data quantities that need to be transferred to the cloud.

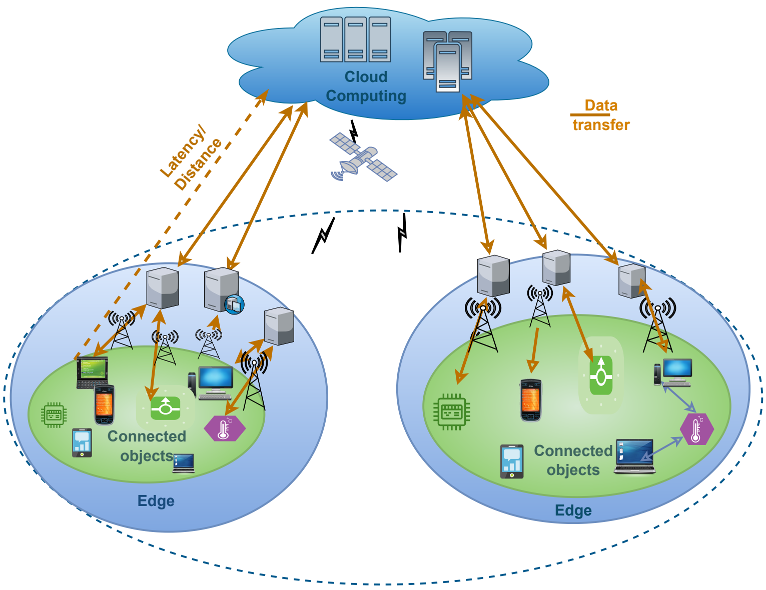

Figure 2.6: Edge Computing approach [1].

2.5.1 Edge computing advantages

Edge computing provides various advantages such as below [69]:

2.5.1.1 Real-time data processing

In edge computing architectures, the computing equipment is placed as close as possible to the data sources, enabling real-time communication. This avoids the recurrent problem of latency encountered with more traditional cloud solutions.

2.5.1.2 Reduced bandwidth

Edge computing promotes local data processing at the Edge gateway. Only data that cannot be processed locally or needs to be brought online is uploaded to the cloud.

2.5.1.3 Increases data security

With an edge computing solution, most of the data remains in the local network. In this situation, companies will find it easier to protect their data and ensure that they comply with compliance requirements.

2.5.1.4 Reduce costs

Edge computing can help reduce costs by enabling nearby resources, eliminating the need to consume remote resources and associated services like storage, data security, and data management policies. By processing data closer to the source, edge computing can also reduce the amount of data that must be transferred over long distances, which can further lower costs.

2.5.1.5 Scalability

In order to add new devices, IoT systems may scale more effectively with edge nodes to meet the workload and expand the user base. This expansion can help user to benefit from IoT services in adequate time. For example, the healthcare domain uses IoT technology to improve health treatment and services. It needs immediate answers about patients' health status to reduce the risk of major medical problems with low latency that ate provided by edge computing.

2.5.2 Edge computing in the healthcare domain

As a result of the growing needs of the healthcare domain for IoT technology, using edge computing has become more important in healthcare systems. It empowers healthcare system efficiency by enabling the real-time collection, processing, and exchange of health data. It provides patients with health-remoting monitoring services that help save their lives with low costs and in short time.

In order to reduce the whole costs of the internet of medical things, Dong et al. proposed a similar network configuration for edge computing in healthcare [70]. Based on the physical boundary of WBANs, the author's structure is divided into two subnetworks: intraWBANs and beyond-WBANs. The proposed system was created to ensure effective resource management and wireless channel resource allocation.

Ghulam Muhammad et al. propose an edge computing-based smart healthcare framework [71]. The framework allows voice disorder assessment and treatment systems. Using a portion of the Saarbrucken Voice Disorder (SVD) database, they achieved 98.5 percent accuracy with the suggested approach. To face COVID-19 spread, Muhammad Usman

	Edge	**Fog**	**Cloud**
Deployment	Distributed	Distributed	Centralized
Components	Physical edge nodes	Physical fog nodes	Virtual resources
Resources	Limited	Limited	Unlimited
Response Time	Fast	Average	Relatively slow
Cost	Relatively low	Relatively Average	High

Table 2.1: Comparing Edge Computing, Fog Computing, and Cloud Computing

Ashraf et al. proposed a model to detect and track contagious individuals [72]. Furthermore, it will use edge computing to preserve the patient's data record for analysis and decision-making. Edge computing also supports the framework that detects and monitors chronic patients. Edge computing can be used to preserve the patients by collecting and process data from edge devices for decision-making.

2.5.3 Edge device

An edge device is any device located at the periphery of a network that generates data. Examples of data sources include sensors and other smart devices commonly found in an IoT environment, such as washing machines, fire detectors, light bulbs, and radiator thermostats.

In the healthcare industry, wearable devices can also function as edge devices by generating and transmitting data. The key characteristic of edge devices is their ability to process and analyze data locally, close to the source of the data, rather than sending it to a centralized server for processing.

2.5.3.1 Wearable devices

The healthcare domain has seen a growing interest in IoT technologies and their potential benefits, leading to a significant increase in the number of IoT medical devices. According to Market Reports, the global IoT medical devices market is expected to grow from USD 26.5 billion in 2021 to over USD 94.2 billion by 2026 [73].

According to Statista [74], it is also estimated that by 2020, more than 161 million of these devices are connected around the world. Wearable devices are among the most popular IoT medical devices and have become increasingly common in recent years. It is a set of devices that can wear by patients as an accessory or attached to clothing, or worn equipped with sensors or small electronics. It can assess the physical and chemical

Wearable Devices(sensor)	Vital signs	Unit of measure	Observation
Glasses (Thermometer)	Body tempe-rature	Celsius or Fahrenheit	- Measures the temperature of the body -The estimation influenced by : Age, .. -Many unit of measure
Bracelet (Blood pressure sensor)	Blood pressure	mmHg	-The estimation influence by : Age, . - Systolic (S):heart contracts - Diastolic (D): heart relaxes
T-shirt (optical sensor)	Heart rate	Beat per minute	-The estimation is influenced by: Age, ambient oxygen,.
Gloves (respiratory rate sensor)	Respiratory rate	Breaths per minute	-The estimation influenced by: Age, ambient oxygen...
Smart watch (photo detector)	Oxygen saturation	percent	- Estimative the amount of oxygen in the blood

Table 2.2: Examples of wearable devices with some vital signs

features of the body by capturing data concerned with activity duration, sleep status, GPS position, physical activity, vital signs, etc.

The wearable device also analyzes, aggregates, and processes physiological data. Wearable devices can provide healthcare professionals with valuable data about patient health and help individuals track their own health and fitness goals. It enables health monitoring of patients in and outside the clinic and enables the detection of any health risks. It ensures early diagnosis, personalization diagnosis, remote patient monitoring, and improved decision-making. Furthermore, Wearable devices are distinguished by low cost, fitness to wear, and provide real-time results. It encourages patients to use it for monitoring and tracking their health.

Many wearable devices that monitor a person's health are being developed, such as smartwatches, helmets, and glasses. Wearable devices based on sensors can detect heart rate abnormalities, blood pressure, body temperature, or glucose levels faster than legacy technologies, such as finger prick glucose testers. Table 2.2 presents an example of some popular devices for health monitoring with their signs of measurement, a unit of measure.

2.6 IoT's major challenges, limitations, and difficulties

The Internet of Things faces many challenges that limit its performance and slow its progress. To address these challenges, the authors have identified and catalogued them [75]. Although using new technologies like cloud, fog, and edge enables handling these challenges, the interoperability challenge is still the most affected. The main challenges are presented below.

2.6.1 Security

The great results achieved by IoT technology in the development and enhancement of numerous sectors motivate many designers and manufacturers to market their products faster without taking the necessary measures to incorporate security. As a result, many IoT devices spread out in different surroundings without protection from attacks [76].

IoT security is the technical area concerned with protecting connected devices and networks on the internet of things. It entails maintaining privacy and confidentiality, access control, end-to-end security, assuring the security of users' data, infrastructure, and IoT devices, and ensuring the availability of services delivered by an IoT ecosystem.

2.6.2 Scalable architecture

Due to the quick advancement made by the IoT domain, the demand for new things to integrate into networks has increased significantly. However, this integration is not always easy to achieve, as each integration requires more addressing, information management, and service management capabilities. This state represents a barrier to advancing the IoT field. It is necessary to provide the internet of things architecture with the ability to allow the addition of new devices or resources seamlessly without affecting the quality of service of the existing IoT.

2.6.3 Energy Efficiency

Large sensors, devices, and machines continue to consume significant amounts of energy, creating demand for adequate energy sources. Ensuring the correct functioning of these devices while minimizing energy consumption is a crucial challenge in the IoT field.

Energy efficiency is one of the most critical IoT development issues and is respected in the research community. Researchers propose many solutions focusing on reducing consumption through authorizing the necessary communication [77] and providing other energy sources without any effect on the climate [78], such as solar, clean, wind, etc.

2.6.4 Interoperability

Usually, the competitive situation of manufacturers allows the improvement of the products and progress in their quality. However, it also leads many manufacturers to create their own devices, define their own protocols and interfaces, and implement their architecture. These issues create new phenomena called "silos"(see Figure 2.7), where the IoT applications cannot exchange data between platforms or domains [2]. This phenomenon also limits the ability to connect IoT devices across different platforms and applications, potentially increasing the cost and limiting overall IoT adoption due to the need for interoperability.

Furthermore, to the vertical silos' phenomenon consequences, the domains are separated where most domains in the real world are not, and the information of each domain may be more important in others. For example, information about the weather domain is considered more important in the agriculture domain, and transportation information is associated with the industrial domain.

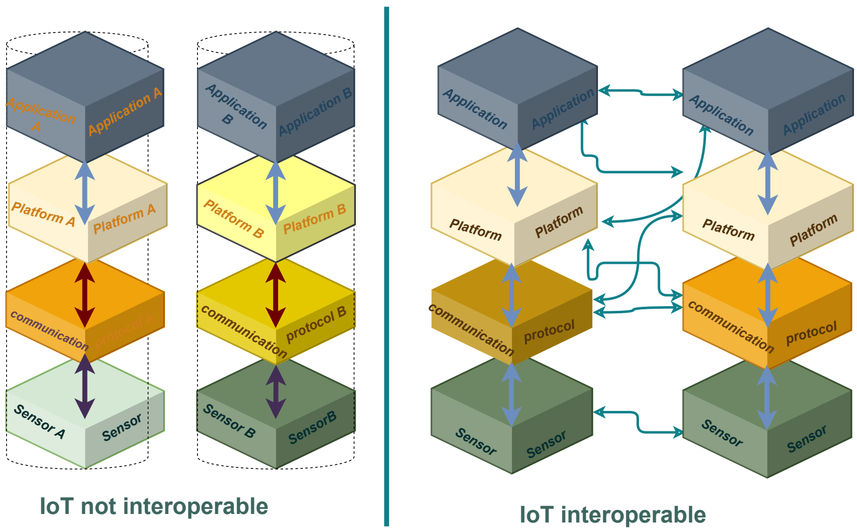

Figure 2.7: Interoperable-noninteroparble [2].

IoT applications enable the treatment of IoT data and interpret them to make decisions, provide services, or exchange with others. Providing an IoT application can not combine the collected domain data from heterogeneous sources or exchange them due to reducing their efficiency and creating a real challenge to interoperability.

2.6.4.1 Definition

The term *"interoperability"* is defined in Cambridge Dictionary as *"the degree to which two items, programs, etc. may be used together, or the quality of being able to be used together"* [79].

Interoperability has a variety of definitions in the literature. It is defined by the IEEE as *"the ability of two or more systems or components to exchange information and use the information exchanged "* [80].

IoT interoperability also enables the addition of a new object or device to an existing network and allows them to transfer data information seamlessly to allow others to process it quickly and easily. Furthermore, interoperability can be described as the ability of various systems, organizations, and/or individuals to collaborate to achieve a common goal.

According to the McKinsey report. *"Interoperability is required for 40% of the entire potential economic value enabled by the internet of things, and nearly 60% in some cases"* [17].

Many vendors focus on the standardization of IoT devices, network material and protocols, and data formats to achieve interoperability between multiple vendors. Standardization is a cost-effective solution and opens up opportunities in the IoT domain, and is supported by the industry. However, the solution necessitates the use of standard protocols and platforms to link devices from multiple providers to communicate with others, which becomes difficult with the increasing vendor numbers.

The interoperability can classify at many levels. As shown in figure 2.8, the authors' Hafizur Rahman et al. in [81], and Antonis Pliatsios in [82]classified the interoperability of IoT in four levels: technical interoperability, syntactic interoperability, semantic interoperability, and organization interoperability with the possibility of dividing interoperability level in two or more level. Mahda Noura et al. [83] also proposed a taxonomy of interoperability for IoT based on four levels that replace the organization interoperability with platform interoperability.

2.6.4.2 Technical interoperability

Technical interoperability consists of device, network, and platform interoperability.

2.6.4.2.1 Device Interoperability

As IoT technology consists of a variety of devices, transmitters, switches, actuators, Raspberry Pi, and smartphones, this makes it very difficult to integrate and adding new devices. Interoperability refers to the capacity of heterogeneous IoT devices to integrate and com-

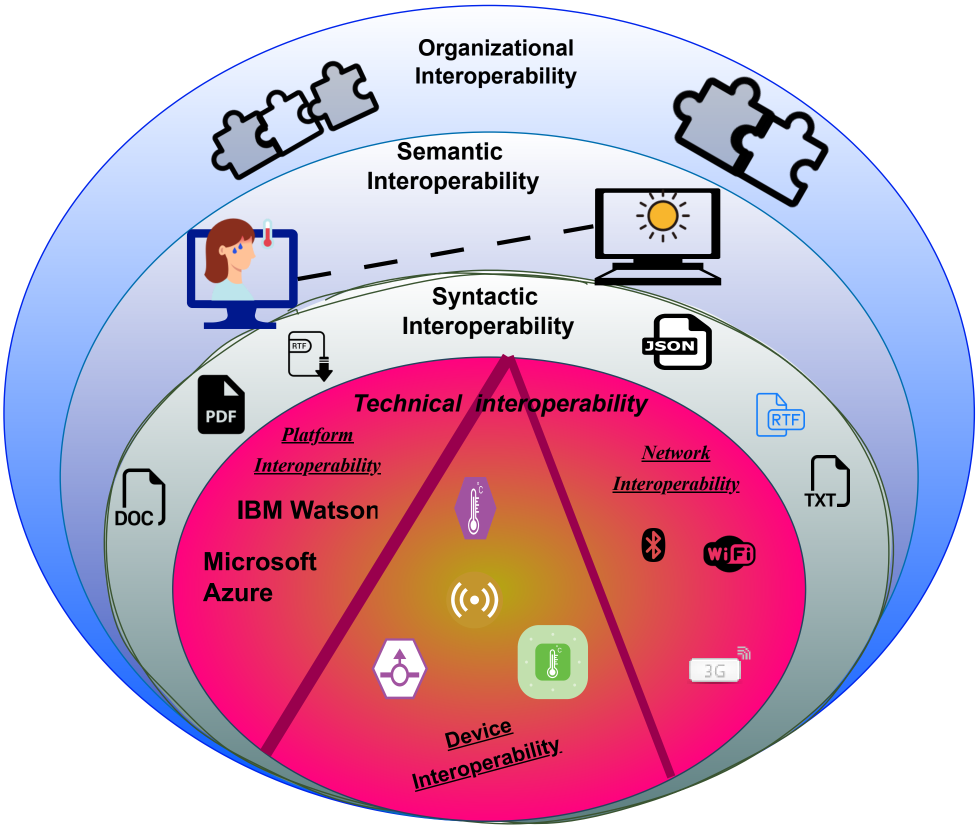

Figure 2.8: IoT interoperability levels.

municate with one other using various communication protocols and standards. It also refers to the capacity to integrate new devices into any IoT platform.

2.6.4.2.2 Network Interoperability

The development of various connection protocols distinguishes the IoT market. It generates heterogeneous networks with Wi-Fi, Bluetooth, ZigBee, 3G, and 4G communication technologies. A continuous flow of communications between systems on these networks is required for proper operation. The mechanisms that enable the continuous exchange of messages between systems on these networks for end-to-end communication are referred to as network interoperability. Network interoperability addresses challenges such as addressing, routing, resource efficiency, security, and mobility support.

2.6.4.2.3 Platform Interoperability

The IoT environment comprises a variety of platforms. Some of the popular platforms are Google Cloud Platform, IBM Watson IoT, ThingWorx, oneM2M, Microsoft Azure Cloud, and ThingSpeak [84]. Many of these platforms are personalized for specific IoT applications; the Interoperability difficulties of IoT platforms arise. To accede to these platforms, the user needs to make an effort to use specific plugins such as API. It creates a barrier that prevents the adoption and integration of the IoT system. Platform interoperability is concerned with achieving the exchange of information between heterogeneous platforms within different domains.

2.6.4.3 Syntactic interoperability

The internet of things domain encompasses various devices that generate data in various formats, including TxT, CSV, XML, JSON, and others. Heterogeneous systems have trouble interpreting and exchanging these data types with other systems. Syntactic interoperability refers to the data structure and coding utilized in any transfer of information between different systems. It allows organizing all data into a standard data format and protocol so that the processing method can be determined from the structure. It also allows receiving systems to detect grammatical errors and requires that any communication that appears distorted or incomplete be retransmitted.

2.6.4.4 Semantic interoperability

The internet of things generally needs to communicate to exchange information and resources in the form of programs, data, or services. As most of the data generated by heterogeneous IoT devices and the exponential of data and resources exchanged between different IoT systems are growing, the degree of information heterogeneity also increases, making it very difficult to understand and process. It gradually becomes more ambiguous and inconsistent to process. For example, the creation of data represented by various units of measurement.

Semantic interoperability concerns processing the meaning of data exchanged between IoT systems to ensure its correct interpretation. It addresses consistency issues in the context that the information exchanged should have the same meaning among these systems. Semantic interoperability also seeks to preserve the semantics of the information exchanged. It refers to the capacity of various services and applications to efficiently exchange information, both online and offline, in a meaningful and understandable manner. [85].

2.6.4.5 Organizational interoperability

Organizational interoperability is the high level of interoperability. It is based on the effectiveness of the lower level (technical, syntactic, and semantic interoperability). It concerns the organization's capacity to exchange data generated with heterogeneous devices through diverse networks across different platforms to provide standard structure, ease its interpretation, and ensure correct meaning.

2.7 Conclusion

This chapter presents an overview of the internet of things technology through its definition, function, and architecture. As we have seen, the internet of things greatly impacts the development of other domains, especially health care. In addition, we have presented cloud, fog, and edge computing which represent real support to the IoT technology as they offer new capabilities and services. As a result of the benefits offered by these paradigms, the users of the domain find that it represents a good choice for using their services.

With all the benefits of IoT technology, there are still limits that we have detailed in this chapter. In particular, we emphasize the challenge of interoperability that reduces the potential of the IoT domain. In the next chapter, we will present a state-of-the-art on the semantic web and the evolution of the IoT to a semantic web of things.

Chapter 3

Where IoT meets Semantic Web Technologies: Exploring the Web of Things and Semantic Web of Things

3.1 Introduction

The increasing demand for IoT technology in various fields has led to an increase in the number of devices that have produced huge data, which in turn requires special processing. Most of this IoT data has been distinguished by its heterogeneity and difficulty in exchanging between IoT applications or among different domains. Furthermore, the data generated represent one domain and cannot use more of it. To handle this problem and achieve interoperability, semantic web technologies are suitable solutions for processing and benefiting from this data. The semantic web is based on a set of standards such as the RDF (Resource Description Framework), RDF Schema (RDFS), OWL (Web Ontology Language), and others. The semantic web is also based on ontology to represent, share, and reuse data. In this chapter, we define the semantic web and explain the semantic web technologies architecture, especially ontology, where we present its role, component, and type. We also present the evolutions of the Internet of things to the Web of things and then to the semantic Web of things to respond to the needs of the IoT domain to meet the challenge of semantic interoperability.

3.2 Web of things

Today, the web has already become the main means of communication on the internet; the term "Web of Things" (WoT) describes the use of web technologies to support interoperability between IoT platforms, communication protocols, and application domains. The web of things is an extension of the internet of things; it provides an application layer that simplifies the use of the internet of things and allows sharing of data around the world [86].

The web of things focused on applying architecture based on the principles of RESTful and REST APIs to take advantage of the popularity and maturity of the web for unifying the cyber-world and physical worlds. W3C supported the WoT with a metadata sequence modelling interaction patterns (properties, actions, and events) called Thing Description [87]. WoT technologies are available today in a number of solutions from W3C members and others in the industry's technology, such as Siemens Desigo CC, Node-RED, Mozilla Web Things, and Fujitsu. Despite the Web of Things' (WoT) efforts to develop a uniform infrastructure for the integration of devices and services using web protocols like RESTful architecture and HTTP protocols (as shown in Figure 3.1), Interoperability is still challenging due to differences in protocols, data formats, and other factors. Furthermore, the WoT also needs to integrate data and establish semantic interoperability in order to reach its full potential, which can be made possible by tools like the Semantic Web.

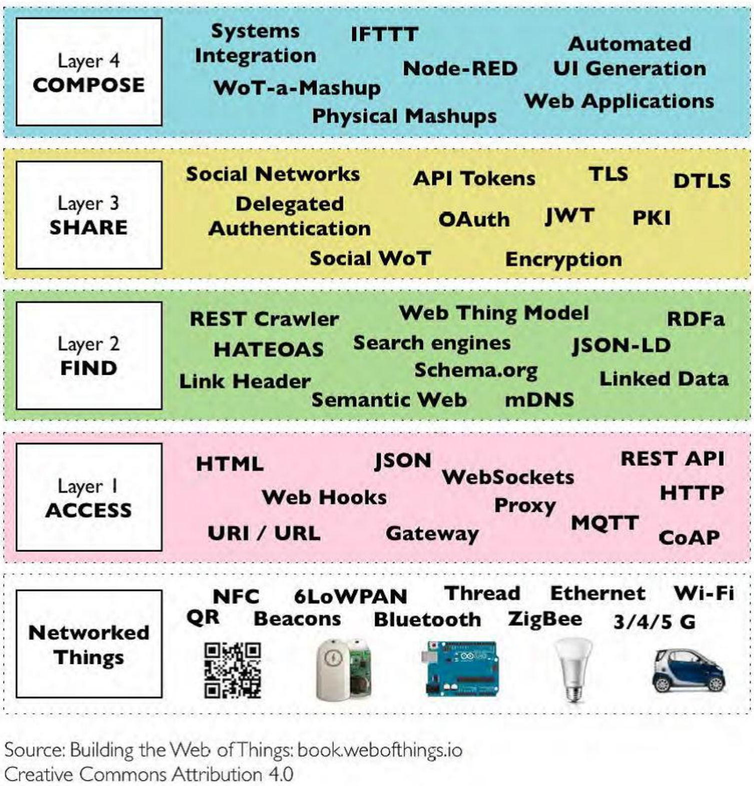

Figure 3.1: Web of things architecture [3]

3.3 Semantic Web

3.3.1 Definition

The Semantic Web is an extension of the World Wide Web Consortium standardized web. Berners-Lee proposed it in 2001 with the goal of making web pages understandable by computers. As its name indicates, the primary ambition of the Semantic Web is to give "meaning" to the information on the web by making the "knowledge" contained in the latter both exploitable and accessible by computers via Web technologies. Berners-Lee defines the Semantic Web as *"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better-enabling computers and people to work in cooperation"* [4]. From this definition, we can determine the objective of the Semantic Web, which gives information a well-defined meaning for helping humans and machines to work together more effectively. The semantic web address this objective by including a set of techniques. The techniques allow data to be specified and linked in such a way that computers may use them not just for display but also for automation, integration, and reuse amongst different applications.

The semantic Web overcomes the classic web limits by providing semantic links, which helps the keyword search engines to avoid leaving much work to the user. It enables us to identify the right resources, analyze the content of the pages, find the right information, combine the different results, etc. The semantic Web focuses on sharing common meanings and machine metadata processing to allow users to manipulate the Web's full potential.

3.3.2 Architecture

The Semantic Web is designed to make the content of World Wide Web resources accessible and useful. The architecture of the Semantic Web, proposed by the W3C, is based on a pyramid representation of languages [4]. Figure 3.2 shows the layered organization of the Semantic Web languages.

3.3.2.1 URI/Unicode

URI and Unicode constitute the base layer of the proposed architecture. URI (Uniform Resource Identifier) enables the identification and localization of various resources. It offers a string of characters that serves as the resource's special identifier. This resolves the issue with natural language, where the same word can have multiple meanings depending on the context. Using different URIs (Uniform Resource Identifiers) is a common way to distinguish different items or concepts on the web. For exam-

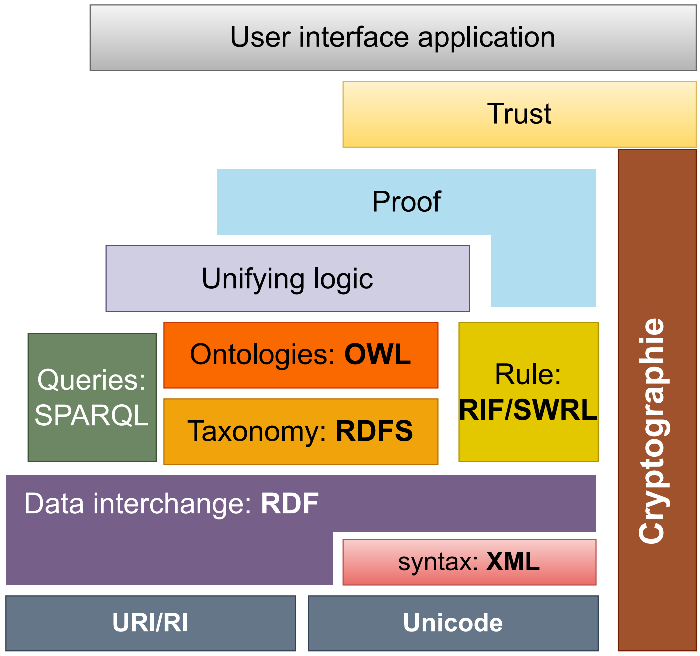

Figure 3.2: Semantic web architecture [4].

ple, the URI https://example.com/fruits represents the concept of fruit, and the URIs https://example.com/fruits/apples and https://example.com/fruits/oranges are used to represent the specific fruits apples and oranges, respectively.

Unicode offers a universal character encoding standard for text and characters. As a result, it is possible to share and develop international software as well as encode words in other languages. Currently, Unicode can represent and encode about 1 million characters, which is sufficient to represent all the world's written languages.

3.3.2.2 XML

XML is a markup language for structuring documents according to their content. It provides a common manner of structuring the document to enable the machine's tasks, such as the researcher, moving, displaying, and updating its content. Different to other languages, the user's XML can define the structure of their document according to their proper tags without taking in compute the environment or system that is used. As XML is a simple language readable by both humans and machines, it has become a default format in many applications such as Microsoft Office and others. XML is currently used as a standard for data transmission on the internet; it is used as the syntactic base layer of the semantic web(see Figure 3.3).

```
<?xml version="1.0" encoding="UTF-8"?>

<patient_information>
<personal_information>

    <name> mohemed amin</name
    <age>45</age>
    <adress>Guelma</adress>
</ personal_information>
<disease> Has diabetes</ disease >
</patient_information>
```

Figure 3.3: Example of XML code.

3.3.2.3 RDF

RDF (Resource Description Framework) is a specification of the World Wide Web Consortium (W3C) family, originally conceived as a standard model for the semantic representation of Web information. It is defined to use metadata for web resources, then become machine-readable and exchange format [88]. The RDF language provides a way to represent information in the most flexible and generic way possible. It is based on a fixed model composed of assertions (declarations) and uses XML's syntax to represent the content RDF is based on the notion of triples, as in the example in Figure 3.4, each triple consisting of a subject naming the resource to be described, a predicate representing the property applied to the resource, and an object representing the property's value. RDF graphs are extremely effective tools for information representation, but a computer can never read them without serialization. The normative syntax for RDF serialization is RDF/XML. Other serialization formats, such as TURTLE and N3 syntaxes, are less verbose than RDF/XML. RDF includes a set of predefined predicates like (rdf:type) that define the belongs of an instance to a class. However, it has a limit to defining a richer constraint, providing more expressively, or making reasoning.

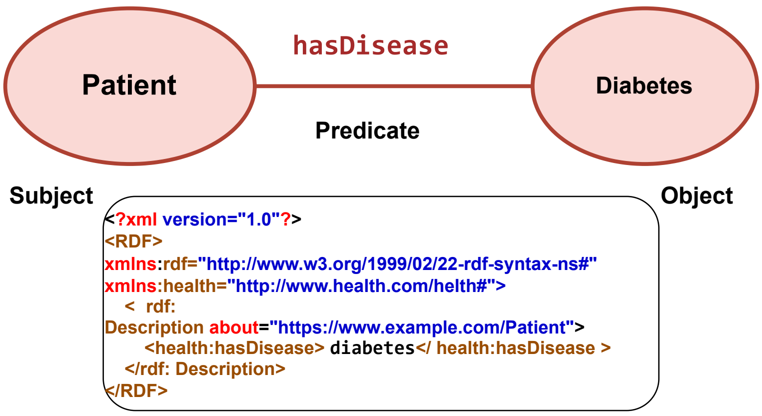

Figure 3.4: RDF triple graphique representation and RDF/XMLrepresentation

3.3.2.4 Rdf schema

In 2004, The W3C published and recommended an extension of the RDF language called the RDF schema. It allows for a higher level of abstraction [89]. The RDFS has added to RDF the ability to define classes or hierarchies of classes (rdfs:Class, rdfs:subClassOf), define a property hierarchy (rdfs:domain, rdfs:range) and also define the constraint on the value that is associated to the properties to provide it meaning. According to the semantics given in RDFS, RDFS expressions are written as RDF triples, in following an example of RDF schema (see Figure 3.5). Although RDFS facilitates data inference and enhances search on that data, unfortunately, it is not expressive enough to handle complex semantics such as:

- Express two classes that are disjoint, for example, classes of female and male.
- Allow representing cardinality in RDF, which allows defining restrictions on the number of occurrences of values that a property can take. For example, we cannot say that a person has exactly two parents.
- Allow creating classes by combining other classes (intersection, union, complement).
- Reason about these representations when the semantics remain very limited.

3.3.2.5 Owl

The W3C recommended the Web Ontology Language (OWL) in 2004 as an RDF-based language that improves the RDF Schema model by defining a rich vocabulary for representing complicated ontologies [90]. OWL is built on a strict syntax that defines formal

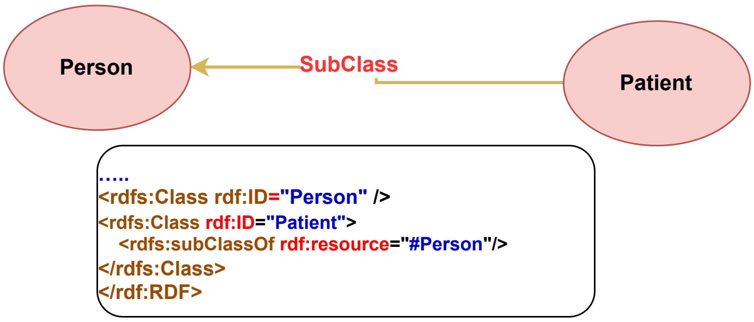

Figure 3.5: RDF Schema

semantics and enables the creation of ontologies that can be subsumed. It is a more comprehensive language that includes properties such as equivalent class, equivalent property, the identity of two resources, differences between two resources, contrary, symmetry, transitivity, and cardinality, among others. Through these properties, It enables the definition of complex resource relationships. It also enables the definition of constraints that verify the data's consistency and helps deduce new knowledge from the data. OWL document content is defined by the RDF triplet with a specific and precise description using a fixed predicate like(owl: SymetricProperty). OWL is available in three versions are depicted in Figure 3.6, OWL Lite, OWL-DL, and OWL Full, which range from the simplest to the most complex:

OWL Lite: The main benefit of this language is that it allows for modelling basic ontologies with a low level of formal complexity, making it straightforward to implement correct and complete reasoning systems.

OWL-DL: This language covers all OWL structures with a few exceptions, such as type separation: a class cannot be both an individual and a property. Because it relates to the description logic, it is known as DL. It is considered adequate for users who desire the greatest degree of expressiveness while maintaining completeness and decidability.

OWL Full: as this language is designed for people who want maximum expressiveness, it includes all the constructors of OWL DL and all RDF Schema constructors. It enables the usage of a class in the constructors in place of an individual.

3.4 Ontology

Ontology is a fundamental concept in the field of semantic web technology. It provides a means for representing data using concepts and their relationships, enabling the reuse and exchange of data between different systems and domains that share a familiar context [91].

Figure 3.6: OWL categories

3.4.1 Origin

The word "ontology" comes from the domain of philosophy; the verbal element ontological comes its constructed from the Greek roots "onto" which means what exists, and "logos" which wants to say the study, hence its translation by "the study of the being", which the philosopher Aristotle defined as the science of existence [92]. Ontology is also interpreted in the domain of artificial intelligence as the whole of what exists, with its relations, restrictions, axioms, and vocabularies. These characteristics should be similar to the real world.

3.4.2 Definition

Many definitions have been given for ontology in computer science: One of the first definitions was given by Neches et al. [93]. They defined ontology as *"An ontology defines the basic terms and relations comprising the vocabulary of a domain, as well as the rules for combining these terms and relations to define extensions of the vocabulary"*.

Gruber also defined ontology as the *"explicit specification of a conceptualization"* [94]. Studer revised this definition to convert: *"An ontology is a formal, explicit specification of a shared conceptualization"* [95]. This definition can be understood in the following way:

The term" Conceptualization" describes the establishment of the abstract model that simplifies the vision of the domain of phenomena it represents.

The term "specifications" is the formal definition of the terms that describe a domain, the relationships between them, and the axioms that constrain them.

The term "Formal" means the ontology is readable and interpreted by a machine.

The term "Explicit" signifies that the type of concepts, the properties, the relations, the constraints, the axioms of the ontology, and their uses are explicitly defined.

The term "Shared" indicates that the ontology takes consensual knowledge (shared by a community).

According to these definitions, an ontology is a formalism that represents the general properties of what exists, enables automatic processing, and is frequently used to express knowledge. To correctly use an ontology, the ontology developer or reader should know all its roles to use it properly.

3.4.3 Role of ontology

The ontology defines a common vocabulary to facilitate sharing information about a domain between individuals, between individuals and systems, and between systems. It includes machine-interpretable definitions of basic domain concepts and their relationships, eliminating the ambiguity of terms and offering inference potential [91]. Some of the reasons that lead to developing ontologies are:

- Sharing a common understanding of the structure of information between systems, as well as between people and machines.
- Facilitating access to heterogeneous resources.
- Standardize vocabulary across a domain and provide a common understanding.
- Allows domain knowledge to be analyzed and relevant data to be extracted.
- Make explicit what is considered implicit in a domain (define classes, relationships, and instances).
- Improving information retrieval processes.
- Ensuring and facilitating the reuse of knowledge concerning a domain (defined once but used in several applications of the same domain) and minimizing the loss of information.
- Ensure the cooperation between different systems of information.

3.4.4 Components of the ontology

Ontologies are used to formalize vocabularies in order to represent knowledge. Ontologies allow defining concepts (classes) as well as attributes (data properties) and relations (object properties) between these concepts. These components play a crucial role in their development, modification, and validation [91].

3.4.4.1 Concept (class)

Represent abstractions of a specific aspect of reality (the domain). It constitutes the concepts that are referred to, which are selected according to the stated objectives and purpose of the ontology. The concept can represent a group of resources with similar characteristics. It can also represent a physical object, a profound notion, an action, a reasoning process or an idea, etc. The concept constitutes the main entity of an ontology. It is also named class, or each class is associated with a group of individuals. The class is defined by a set of properties and relations that link it with the classes that surround it. A concept(class) is an entity composed of three different elements:

1. The term is used to express the concept in language.

2. The concept's meaning, also known as its "notion" or "intention".

3. The thing(s) the notion refers to, also known as the "realization" or "extension" of the concept.

3.4.4.2 Relationships (Property)

The properties are used to define how classes and individuals can be related to each other in the domain. Properties are non-predefined, non-taxonomic relationships. It exemplifies the added utility of ontologies that are able to describe various particular non-predefined relationships. The property consists of two main types:

- The first type of property is called an "object property". It is specified in such a way that the first argument of the relation corresponds to the domain of the property and the second argument to its co-domain (a concept related to the domain by the property). As a result, it describes a relationship between two individuals or instances.

- The second type of property, data property, it establishes a connection between a class of individuals and a data value (data type). The latter is expressed the characteristics of concepts, which serves as the co-domain in a relationship, where the domain is a concept.

Both previous types of properties can be arranged hierarchically and specify in which direction the object is related to using properties like Subclass-of (generalization/specialization); Part-of (aggregate or composition); Associated-with; Instance-of. It can also use predefined conceptual relations like equivalence, disjunction, and many more can be used to connect them.

3.4.4.3 Individuals

The individuals of an ontology can include concrete objects such as people, animals, countries, etc. They are examples of a class. They inherit all the properties of that class. These specific objects were created by concepts using the instantiation relation "Instance of" or "is kind of" or "type", which was predefined. An ontology need not include all individuals, but one of the general purposes of an ontology is to provide a way to classify individuals, even if they are not explicitly part of the ontology.

3.4.4.4 Axioms

The axioms are used to specify the ontological assertions that will subsequently be accepted as true. This determination describes the significance of ontology components, relation arguments, and attribute value restrictions. They are also used to verify the ontology's consistency and enable using a "reasoner" to deduce knowledge that is not expressly expressed. Ontologies' explicit axioms hold a greater amount of implicit knowledge that can be deduced by the reasoner the more sophisticated they are.

3.4.5 Ontologies type

The literature has many kinds of ontologies based on proposed specific criteria. The degree of abstraction and application domain is the most frequently used criteria. Guarino [96] proposed the following classifications based on their degrees of generality and dependence:

3.4.5.1 Top-level ontology

It is also called superior or basic ontology and used to represent common or consensual knowledge, including vocabulary related to things, events, space, time, causality, and function. Since this type of ontology describes general and problem-independent concepts, they are considered an ontological core that can be reused in other domains. Top-level ontology represents ontologies with concepts common to many knowledge domains.

3.4.5.2 Domain ontology

This type of ontology is highly advised. It supports reusing in several applications of the same domain. It provides a good expression of vocabulary associated with a generic domain. Its goal is to describe a specific topic of knowledge, which may be broad or more narrowly focused and only represent a portion of a knowledge domain [96] .

3.4.5.3 Task ontology

This ontology's concepts are more specific to the generic tasks of an application or a process. They are also acceptable for usage in domain ontologies because they explicitly conceptualize formalism and model the primitives used in the knowledge [96] .

3.4.5.4 Application ontology

The concepts utilized in this ontology are tailored to a particular field and cannot be generalized to other contexts [96] .

3.4.6 Correspondence between ontologies

To ensure data sharing between heterogeneous knowledge bases and the reuse of information from these bases as well as to minimize information gaps, cooperation between ontologies through the application of certain techniques is necessary. Ontology alignment and merging represent some of these techniques.

3.4.6.1 Alignment

Numerous ontologies have been defined to provide a formal representation of knowledge of many domains, and some of them represent similar knowledge. Making the correspondence between ontological concepts is the process that allows the linkages between them [97].

3.4.6.2 Merging

Ontology merging refers to the process of creating a new ontology from several source ontologies without including any further information. As a result, there must be a common subject throughout all ontologies [97].

3.4.7 Ontology construction methodology

In order to have a sustainable ontology that satisfies the needs, following and respecting the rules of construction of ontology is the assurance manner. In the absence of correct or typical methods for constructing ontologies, many ontologies methodologies are proposed to help in the development of ontologies. Most of these approaches represented an evolution of their previous methodologies that are still used today. METHONTOLOGY [5], ON-TO-KNOWLEDGE [98], and DILIGENT [99] are a set of methodologies that are more usable. They support ontology developers in designing a consistent, maintainability, and quality ontology by offering instructions and describing an iterative approach to ontology development.

The methodologies process started with making the first draft of the ontology, then revising and refining the evolving ontology and filling in the details. Most new methodologies have based on five steps defined by the Methontology [5](see Figure 3.7).

3.4.7.1 Methontology steps

3.4.7.1.1 Specification of the requirements: The specification phase aims to produce an informal, semi-formal, or formal ontology specification document utilizing a collection of intermediate representations or competence questions. It enables the surrounding range of ontology studies by explaining the ontology's goal, including its intended applications, use cases, end users, etc. It also defines the scope, which entails the qualities and granularity of the set of terms to be represented.

3.4.7.1.2 Conceptualization: Structures domain knowledge into significant models through expressing the issue and the solution in terms of the domain vocabulary. It consists of constructing a comprehensive glossary, including concepts, instances, verbs, properties and data dictionaries, attributes, constants, instances, conditions, and rules.

3.4.7.1.3 Formalization: Refers to converting the conceptualization acquired from the previous phase and domain experts may understand into a formal or semi-computable model. It is done through the use of a formal language or formalism, which consists of a collection of semantic components (content), structural rules (instructions for use), and a specific formal notation (form) designed to arrange the relationships between the ontology's constituent elements. Different formalism such as description logics [100], and semantic networks [101] can be used to formally represent an ontology.

3.4.7.1.4 Implementation: Converting the formal model to the ontology suitable for use when it can be populated and queried. The implementation is based on employing some practical technology, such as a programming language, an ontology editor, or an ontology creation programming library. The implementation allows a machine to manipulate domain knowledge via this ontology.

3.4.7.1.5 Maintenance: Even though the earlier stages of ontology development were effective, there may still be some flaws or mistakes that need to be fixed. The ontology's quality will be raised during the maintenance phases through verification, rectification, and validation. Verification involves determining whether the ontology is correct. In validation, we determine whether the ontology achieves its goals within its constraints.

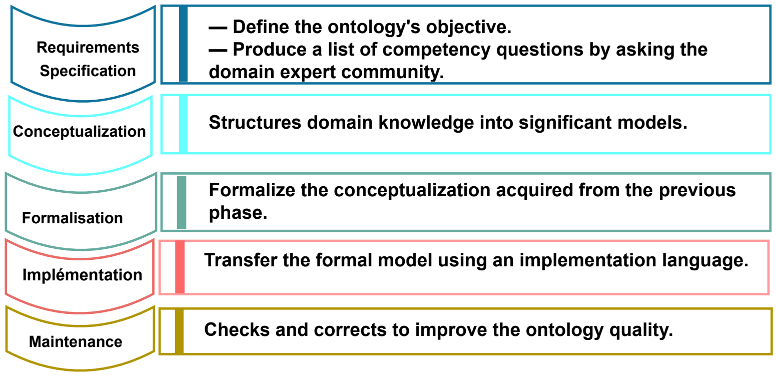

Figure 3.7: Ontology construction methodology [5]

Although these methodologies represent a guideline for building an ontology based on its iterative process, they are more aimed toward ontology researchers. They are adequate for single ontology development and cannot ensure consistency and shared ontology development. They also have limits to obtaining an ontology extension or offering the reusing of other ontologies. The methodology101 is one of those that encourage the reuse of ontologies. The seven steps described by Noy and McGuinness comprise this technique, which is clearly stated in [102]. It suggested starting the development by determining the domain and scope of the ontology step, then reusing existing ontologies, enumerating important terms, defining classes and class hierarchy, defining properties of classes, defining properties of properties, and creating instances. However, this methodology does not include an evaluation and validation step.

3.4.7.2 Neon methodology

The Neon methodology proposes an ontology construction framework broader than the five ontology development models by describing a set of nine scenarios for ontology design. The Neon methodology is based on some guiding principles [6]:

- The methodology should be general to avoid being used to solve particular circumstances and independent of the available technology.
- The methodology explicitly identifies each process or activity, lists all the methods, techniques and tools that will be used, and specifies its purpose, inputs, and outputs.
- The methodology allows software developers and ontology experts to adopt it quickly.

The NeOn Methodology addresses the limitations of the previous methodologies while benefiting from their advantages. It has kept the phases that are mentioned above, such as the specification, the formalization, the conceptualization, and the implementation. The NeOn Methodology considers features related to ontology development, such as the existence of multiple ontologies in ontology networks, cooperative ontology development, the dynamic dimension, etc. The methodology consists of nine scenarios for building ontologies are presented in Figure 3.8. In the following scenarios:

Scenario1: "building ontology networks from scratch". This scenario represents the state of absent existing knowledge resources to reuse in buildings the ontology. The developer should follow the previous phases applied from the specification to the implementation phase. Furthermore, this scenario includes the scheduling phase. It identifies the activities and processes performed during the ontology development, their arrangement, and the time and resources needed for their completion.

Scenario2: "building ontology networks by reusing and reengineering non-ontological resources ". This Scenario allows using the no ontological sources and transferring to ontology using a reengineering process.

Scenario3: "building ontology networks by reusing ontological resources ". This scenario applies to ontologies resources that share the same specifications and domains. The reusing depends on the developer's needs, maybe through reusing the whole ontology or part of it.

Scenario4: "reusing and reengineering ontological resources ". This scenario applies to reusing existing ontological resources and using the reengineering process before integrating these ontologies. Advanced ontology resource engineering, ontology resource restructuring, and reverse ontology resource engineering constitute the ontology resource reengineering process. Depending on the requirements of each specific situation, these activities can be carried out at four different levels: the specification level, the conceptualization level, the formalization level, and the implementation level.

Scenario5: "reusing and merging ontological resources ". This scenario is appropriate to state the existing ontologies covering the same domain. The ontology result represents the reusing and merging of these ontologies.

Scenario6: "reusing, merging and reengineering ontological resources ". In this scenario, the set of processes is identical to that of Scenario 5; however, the ontology developers have the opportunity to re-engineer the combined collection of ontological resources instead of using it as is. After reengineering the combined set of ontology resources, the process result should be included in the relevant activity of Scenario 1.

Scenario7: "reusing ontology design patterns". To ensure the accuracy of modeling decisions, to speed up the modeling process, or to reduce the complexity of the modeling.

Access to ODPs allows ontology developers to reuse it.
Scenario8: "ontological restructuring resources ". Modularization of the ontology into different ontology modules, pruning branches of the taxonomy that are considered unnecessary, extending the ontology by adding new concepts and relationships, specifying branches that require greater precision, and adding more specialized concepts and relationships. These processes allow the ontology developer to restructure ontological resources to combine and build an ontology.
Scenario9: "Localizing ontological resources". In order to make the ontology a multilingual ontology, i.e., adapted to a certain natural language, all labels of the ontology, after being conceptualized, must be translated into one or more natural languages as part of this adaptation.

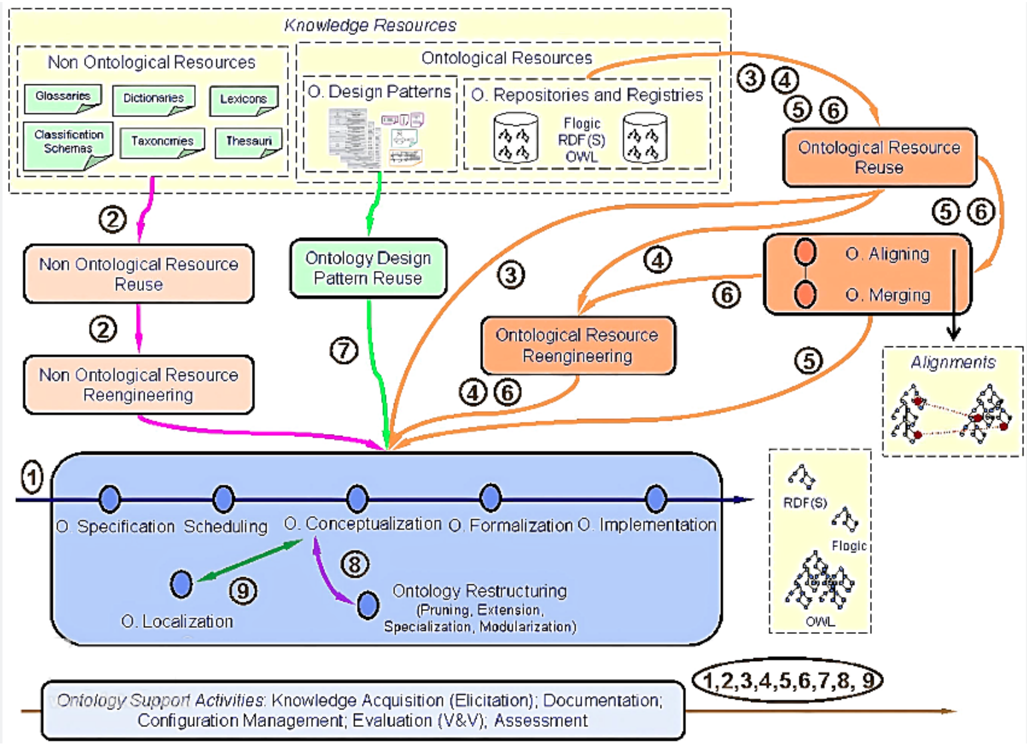

Figure 3.8: Scenarios for Building Ontology Networks [6]

3.4.8 Ontology Editor

Many editors exist to enable ontology creation, modification, and validation. The most popular of them are the following:

3.4.8.1 Protégé 2000

The Stanford University of Medical Informatics offers the free ontology editor Protégé [103]. It enables the creation of an ontology for a specific domain, the definition of data entry forms, and the acquisition of data in the form of instances of the ontology using these forms.

In addition to being an expandable platform, Protégé features a plug-in framework that enables users to manage multimedia material, query, assess, and combine ontologies, among other things. A graphical user interface (GUI) on the Protégé tool enables manipulation of all ontology elements, including class, property, instance, and others. Any domain where concepts may be modelled in a hierarchy of classes can use Protégé.

Additionally, Protégé enables the creation or import of ontologies expressed in a variety of ontology languages, including RDF-Schema, OWL, etc. It also supports using plugins, which can be downloaded from the internet for different uses.

3.4.8.2 NtoStudio

The editor is developed by the company Ontoprise based on IBM Eclipse [104]. It allows the creation and maintenance of ontologies in different formats by using graphical means and tools for ontology modeling tools and rules to integrate heterogeneous data resources. In order to expand its functionality and customize it to the demands, it permits the addition of additional self-developed modules. It stands out for its broad range of modeling capabilities for simple ontologies.

3.4.8.3 OntoEdit

The editor is a flexible and extensible ontology engineering environment based on a modular architecture based. It shares many aspects with the previous editors. Due to their extensive functionality, it is proposed to edit, browse, and maintain ontologies and import and export ontologies in various formats (XML, RDFS, etc.) [105].

3.5 SPARQL

SPARQL is a query language that allows the exploitation of the semantic representation of RDF data. It provides both a language and a protocol for querying data modelled in RDF. The W3C Consortium develops and recommends the use of the SPARQL language, which can be considered the SQL of the Semantic Web; it uses the concept of identifying paths in a graph to retrieve the results of a defined query [106]. Thus, a SPARQL query is composed of an identifier (which defines the type of query), an operator (which identifies the corresponding graphs), and modifiers (for example, ORDER BY). A query can query one or more RDF documents, and it can do so using either a FROM attribute at the start of the query or APIs that allow many sources to be examined simultaneously time. The SPARQL query consists of two phases: the declaration phase to represent the namespace in turtle format and the manipulation of the RDF through two fixed commands like SELECT and WHERE and others optional. Figure 3.9 presents an example of a SPARQL query.

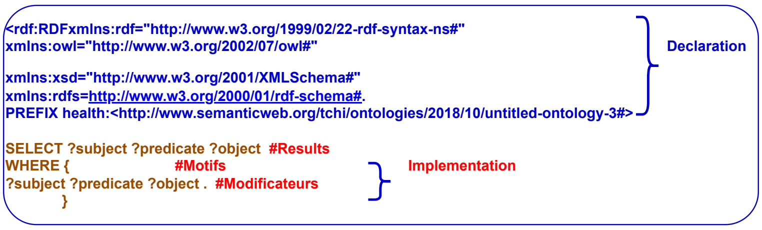

Figure 3.9: Standard Form of SPARQL Query

3.6 SWRL

Semantic Web Rule Language is a W3C-recommended language for OWL expression rules. It allows the creation of rules in terms of OWL concepts and can provide reasoning about OWL individuals [107]. The rules are expressed as an implication between an antecedent and a consequence. Because SWRL permits variables (?x,?y,?z) to manipulate instances, the antecedents, and consequent are formed by the conjunctions of these variables or by the OWL relations between these variables. SWRL does not only allow the creation of concepts or relations, but also the addition of relations according to the variables' values and the rule's satisfaction.

Example: According to the next rule, the uncle property is implied by the composition of the parent and brother properties.
hasParent(?x, ?y) , hasBrother(?y, ?z) -> hasUncle(?x, ?z) rules and permit the creation of custom SWRL built-ins.

3.7 Semantic reasoner

It is a piece of software that is able to deduce logical conclusions from a set of asserted facts or axioms. It is a crucial tool for working with OWL ontologies [108]. It also should always be used before querying an OWL ontology that may the results be associated with not explicit information in an ontology, and need to reasoner to infer this implicit knowledge. Each reasoner's capabilities depend on the axioms and inference rules it knows about, which are related to a particular kind of logic. Several OWL reasoners have been created to apply SWRL rules and deduce novel information, such as FaCT++, Pellet, and HerMiT. HerMiT and Pellet provide more direct support for OWL and SWRL rules and permit the creation of custom SWRL built-ins. HerMiT is the first OWL reasoner that is freely accessible [109]. It is built on a brand-new "hypertableau" calculus that offers far more efficient reasoning than any existing technique. Pellet reasoner enables the definition of unique SWRL built-ins and has more direct functionality for working with OWL and SWRL rules. Pellet considers these rules and provides a conclusion based on them when used to reason over an ontology using SWRL rules. Based on semantic web technology, researchers have shifted their focus from using the web to achieving interoperability between diverse IoT applications towards employing the semantic web as an alternative solution.

3.8 Semantic Web of Things

3.8.1 Definition

Integrating Semantic Web techniques with the IoT leads to a new paradigm, the "Semantic Web of Things (SWoT) ". The SWoT is a recent research field. Incorporating semantics into the IoT domain and ensuring a common understanding of data represents a progression of the Web of Things. The semantic Web of things emerged to handle many challenges facing the IoT domain of development. It enables the heterogeneity treatment and interpretation of data that allows the overcoming miss of standardization problems and provides more potential to the IoT domain. SWoT was first introduced as part of the Spitfire project in 2011. The Spitfire project [110] proposes creating the "Semantic Web of Things" by integrating semantic web technology and the Internet of Things. The Semantic Web of Things has emerged with the objective of:

- Overcome the limitation of the existing WoT platform in handling heterogeneous protocols and enables sharing of sensor data over the World Wide Web.
- Make the data collected more richness by combining data from different physical worlds.
- Infer new knowledge through semantic reasoning and facilitate its retrieval.

- Provide a new form of meaningful content that humans and computers can process.
- Ensure interoperability of IoT applications by allowing machines to interpret IoT data unambiguously.
- Support the discovery process of sensors and make them easily accessible in the physical world.
- The ability to reuse sensor data in the development of new applications.

Ontology is among the most widely used semantic web technologies for managing data heterogeneity and achieving interoperability. In the IoT domain, many ontologies are defined by researchers to address specific problems, which raises the problem of standardization of ontologies.

3.8.2 Internet of things ontologies

In recent years, several sensor ontologies have been proposed in the context of IoT. One of the first ontology is sensor node ontology, designed by Avancha et al. [111]. To define the conditions and expected behaviour of the sensor network, OntoSensor [112] ontology is a generic sensor knowledge base on SensorML, used for querying and reasoning. After that, several ontologies of the IoT domain has proposed. In the following, the most popular ones will be mentioned accordingly to the period of definitions:

CSIRO [113] is a general ontology for the sensor used to describe the composition of sensors and represent virtual sensors. It's designed to make integrating, searching, and categorizing sensors and sensor data easier. The ontology is arranged around four primary clusters of concepts that describe the detection domain (Feature), the sensor (Sensor), the physical components and position of the sensor (SensorGrounding), and the functions and processes of the sensor (OperationModel and Process). Despite these ontologies not being complete and discontinuous, they helped to build new ontologies with other standards for developing use cases. In 2012, the community of the World Wide Web Consortium (W3C) produced the Semantic Sensor Network (SSN) ontology [114]to describe sensors and observations in terms of capabilities, measurement processes, observations, and the sensing method of sensors. The ontology covers a significant portion of the open geospatial consortium (OGC) standards at the sensors and observations levels. The SSN is based on the design pattern of the stimulus sensor observation ontology, with the addition of the DOLCE-UltraLite (DUL) top ontology alignment. Although The SSN ontology had difficulty in describing the actuator concept and the different sensor capabilities, the ontology became a standard in the field of the semantic sensor network by describing core concepts common to all IoT applications. The authors Xiang Wang et al., in a study [115] through the existence of SSN ontology, classified the ontologies into two classes, before the SSN and after the SSN, when most of them are SSN ontology extensions.

The Wireless Semantic Sensor Network ontology (WSSN) [116] enriches the SSN ontology by adding three new concepts: communication, data flow, and state. The ontology describes the context and communication policy of nodes. It adapts the communication of sensor Network ontology nodes to their context to improve the lifetime of the sensor Network ontology.

Bermudez-Edo et al [117]. propose the IoT-Lite ontology as an instantiation of SSN ontology in order to improve the use of SSN in real-world applications. The IoT-Lite focuses on key IoT concepts to avoid the unnecessary concepts and relationships that make the ontology heavyweight and complex to query. The ontology doesn't take into account the performance of data and several devices. The idea of lightweight information models in the IoT domain continues by proposing new ontologies like SOSA.

SOSA (Sensor, Observation, Sample, and Actuator) [118] is a new extension of the SSN ontology developed by the first joint working group of the Open Geospatial Consortium and the W3C. The ontology provides a flexible framework that can include actuators and actuation. It provides a vocabulary for safely exchanging data across all uses of the SSN and its modules. SOSA is easy to use and aligns easily with other standards. It enables the connection with Schema.org to describe IoT devices and their capabilities. Elsaleh T et al. extend. The SOSA ontology to a lightweight semantic model for stream data named IoT-Stream [119].

FIESTA-IoT [120] is a unified ontology that combines several reference ontologies already in use to federate and enable interoperability for Internet of Things (IoT) devices and sensor data. Despite being a unified ontology created by combining multiple reference ontologies into one, FIESTA-IoT has challenging to scale due to its lack of modularity.

Nicolas Seydoux et al., toward the unification of IoT ontologies, propose IoT-O ontology [121]. Their ontology is IoT-O is a federating ontology and gathers many concepts defined in recognized ontologies. IoT-O reuses existing ontologies to provide dynamic knowledge representations. It is based on SSN ontology for sensor definition and the SAN ontology to represent actuator concepts; furthermore, it enables adaption over Time to change the state of the world. The design of IoT-O follows the NeON methodology.

M3 Ontology (Machine-to-Machine Measurement ontology) is an extension of the Semantic Sensor Networks(SSN) ontology [122]. M3 is defined to allow the aggregation and enhancement of cross-domain data in the IoT domain. The M3 ontology describes the main elements of the M2M architecture and semantically annotates M2M data. This ontology integrates M3 concepts to domain ontologies and organizes M2M devices, their data, and domains to better the understanding.

The popular OneM2M [123] platform provided an" oneM2M ontology" ontology. OneM2M ontology allows the integration of semantic metadata to describe the different resources registered in the platform. The ontology defines concepts are high-level concepts such as Objects or Services and are linked to the resources of the oneM2M architecture. It enabled the alignment, for example, the alignment with SAREF. The ontology also provided resource support to other systems to describe their data model by deriving concepts. However, oneM2M is a minimal ontology that represents the curial concepts requiring more concepts to represent a device and its functionality.

In 2015, ETSI published the Smart Appliances Reference (SAREF) ontology for appliances relevant to energy efficiency [124]. SAREF was developed with the goal of connecting various platforms and facilitating data exchange via various protocols. The new ontology has become a standard ontology that can provide a shared model for consensus mapping of existing assets. The SAREF ontology provides building blocks that allow the reuse of different parts of the ontology according to specific requirements. Similar to the SSN ontology, SAREF extended to cover other domains such as SAREF for energy, SAREF for the environment, and SAREF4WEAR for wearable devices. The SAREF ontology supports direct mapping to the oneM2M base ontology. According

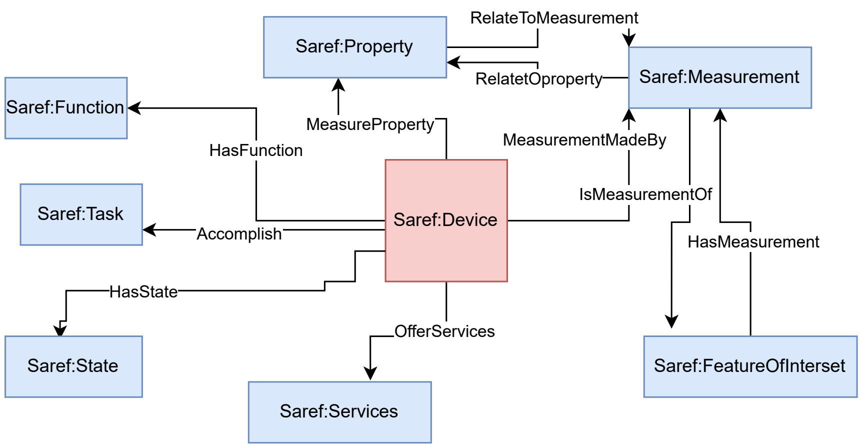

Figure 3.10: Sample of SAREF ontology concepts with relationships(Device) [7]

to Figure 3.10, the main concept of this ontology is saref:Device concept that consists of sensors, actuators, and other devices. As the device accomplishes tasks, functions, and provides services, the three concepts are described and associated with it by the respective relationships *saref:accomplishes, saref:hasFunction, saref:isOfferedBy*. Saref also describes other important concepts of IoT domain like *saref:Measurement* that responsible to represent a measurement value and associated with *(saref:hasValue)*, the unit of measure *(saref:UnitOfMeasure)* and relates to measurement with*(saref:isMeasuredIn)*, and the property *(saref:Property)* represent the property of measurement and devices concepts.

3.9 Conclusion

In this chapter, we have exposed the notion of web semantics as a notion that makes the machine more understandable. We present its architecture that allows the structuring of the data in a standard format and facilitates the definition of the different relations that exist between the knowledge concepts. We have also presented the pillar of semantics called ontology with its origin, definition, role, and components. We explained the existing constructive methodologies and detailed the Neon methodology and its efficiency.

For spreading various format data and the communication protocol, we present the web of things as an integration of the web with the internet of things domain to achieve the standardization problem in IoT. We also present the semantic web of things as a new paradigm allowing us to treat heterogeneous data in web semantics and share knowledge based on ontology specifications. We explain in detail the standard ontologies that have been defined for the IoT domain. In the next chapter, we will discuss the fuzzy ontology to enable this ontology to represent imprecise and vague IoT data using fuzzy logic theory.

Chapter 4

Fuzzy ontology

4.1 Introduction

The Semantic Web of Things refers to the integration of Semantic Web technologies into the Internet of Things (IoT) with the aim of improving interoperability, reasoning capabilities, and decision-making. At the core of SWoT is ontology, a formal and explicit specification of a shared domain conceptualization. Ontology plays a crucial role in representing IoT data, facilitating sharing of knowledge and meaning across different applications and domains. However, it has a limitation in its structure, since it can only handle propositions that are either true or false. This limitation restricts its applicability in situations where vagueness and imprecision in the data. The vagueness problem arises when a value falls on the edge of two categories or concepts. For example, some people are on the borderline of being tall. They are not obviously tall and not obviously not tall.

Furthermore, This limitation becomes more significant when using ontologies in healthcare, as it can lead to inaccurate results and incorrect decisions. Alternative approaches have been developed to address this limitation and improve machine reasoning and decision-making when dealing with vague data, such as fuzzy logic, which provides for the "gradual" characterization of the elements.

Fuzzy logic solves the representation problem of vague and imprecise knowledge. It supports complementing the semantic web of things, especially ontology, for providing a more flexible and accurate way of representing and reasoning vague concepts and relationships.

4.2 Fuzzy logic

The term "fuzzy logic" was originally introduced in 1965 at UC Berkeley by Lotfi Zadeh as an extension of Boolean logic based on the mathematical theory of fuzzy sets [125]. Zadeh found that classical logic could not deal with data that was characterized by vagnuess and imprecision. He created this logic to allow a proposition to be in a state other than true or false. Fuzzy logic is based on fuzzy set theory when a fuzzy set represents a set object that may belong to a certain degree. More formally, a fuzzy set T can be represented by formulating:
$T = (x, \mu_T(x))$; $x \in X$, $\mu_T(x) \in [0, 1]$, where μ_T is a membership function that expresses each element x of the universe of discourse X by a membership value interval [0, 1]. According to Lotfi Zadeh's definition, Fuzzy logic is concerned with the formal principles of approximate reasoning to overcome the precise reason that is considered limiting. Fuzzy logic seeks to simulate the imprecise method of reasoning, which is humans' exceptional capacity to make decisions in the face of ambiguity and imprecision [126]. In this context, fuzzy logic resembles human reasoning like the doctor; for example, when monitoring the body temperature and evaluating the patient's condition, using terms such as "high", "medium" or "low" to provide the decision-making. A fuzzy set is defined by a particular value of its "linguistic variable" and its "membership function".

4.2.1 Linguistic variable

The term "linguistic variable" refers to the idea utilized when describing a circumstance, an event, a process, the pace, temperature, or age. The values of a linguistic variable called "linguistic terms" are the linguistic translations of these distinct states expressed in language [127]. The linguistic variables are used in normal daily activities. For instance, the terms for the linguistic variable temperature include low, medium, and high To determine the degree of membership of an element as a member of a given set, the theory of fuzzy sets has introduced the notion of membership function.

4.2.2 Membership Function

A fuzzy set differs from a classical one in that it is not defined in a way that divides an individual inside a given discourse universe into members and non-members. It is providing each potential individual in the discourse universe a value representing the degree of its membership in the fuzzy set. As shown in Figure 4.1, it is described as a graphical representation of the level of participation for each entry. It allows assigning a value to each processed input, helps in determining the functional overlap between the inputs, and contributes to determining the response of the outputs. The degree of membership to a fuzzy set is represented by a number varying between 0 and 1. Fuzzy logic uses the degrees of membership to determine a given set. The user can choose membership functions randomly or based on their prior experience (membership function chosen by users upon their experiences) and can be created via machine learning techniques (artificial neural networks, k-means, genetic algorithms, etc.). In fuzzy logic, several types of membership

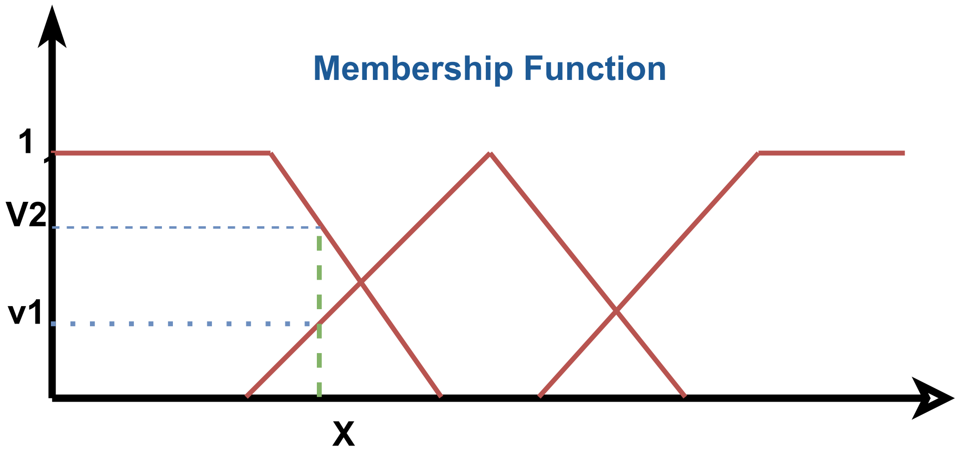

Figure 4.1: Fuzzy logic membership function

functions are proposed in the literature [128]. The selection of the exact type depends on the actual applications. For instance, a Gaussian type should be chosen for systems that require extremely high control accuracy. Thus, the most commonly used membership functions are:

4.2.2.1 Triangular membership function

The triangular membership function is one of the most common membership functions used. It is defined by three parameters a, b, and c. The parameters a and c define the base where the membership degree is zero, and b defines the triangle's height where the membership degree is one [125]. The Triangular membership function is represented by the following formulate(Equation(4.1)) and figure 4.2.

$$\mu(x) = \begin{cases} 0 & \text{x} \quad \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 1 & x \geq b \end{cases} \tag{4.1}$$

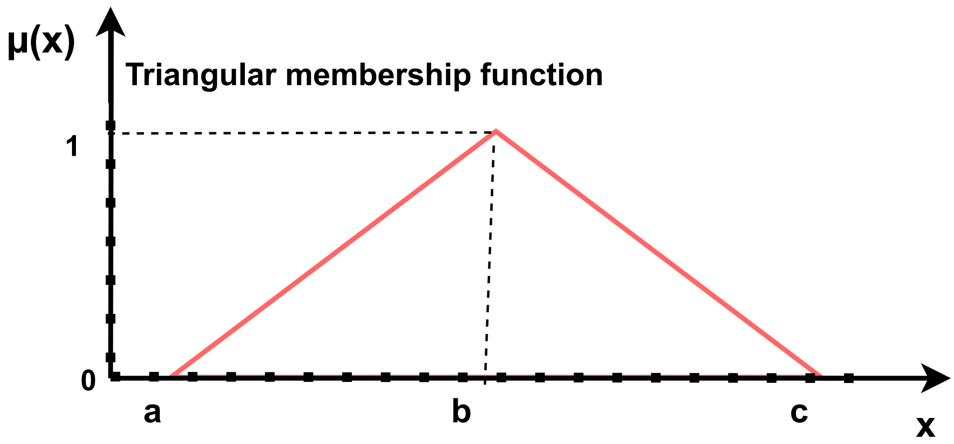

Figure 4.2: Triangular membership function

4.2.2.2 Trapezoidal membership function

By using four parameters, a, b, c, and d, the Trapezoidal membership function is defined(see Figure 4.3). The distance limited by parameters b to c represents the highest membership value that an element can take [127]. The elements between (a, b) or (c, d) can take a membership value between 0 and 1(see Equation(4.2)).

$$\mu(x) = \begin{cases} 0 & \text{x} \quad \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x \end{cases} \tag{4.2}$$

left-shoulder and right-shoulder functions are an extension of trapezoidal membership function where in left-shoulder parameters a, b represented by c, d(see equation 4.3) and with right-shoulder the parameters a,b are the same in trapezoidal(see figure 4.4, equation 4.4)).

$$LeftShoulder\mu(x) = \begin{cases} 0 & \text{x} \quad \geq a \\ \frac{b-x}{b-a} & a \leq x \leq b \\ 1 & x \geq b \end{cases} \tag{4.3}$$

$$RightShoulder\mu(x) = \begin{cases} 0 & \text{x} \quad \geq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x \geq b \end{cases} \tag{4.4}$$

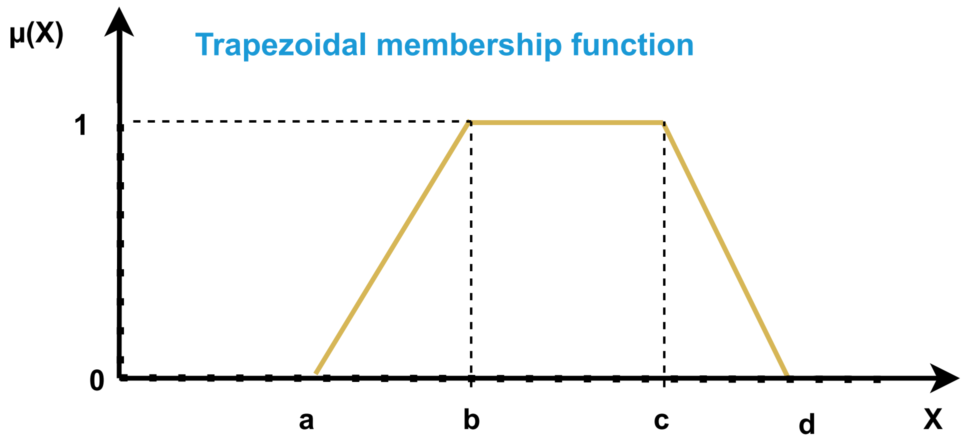

Figure 4.3: Trapezoidal membership function

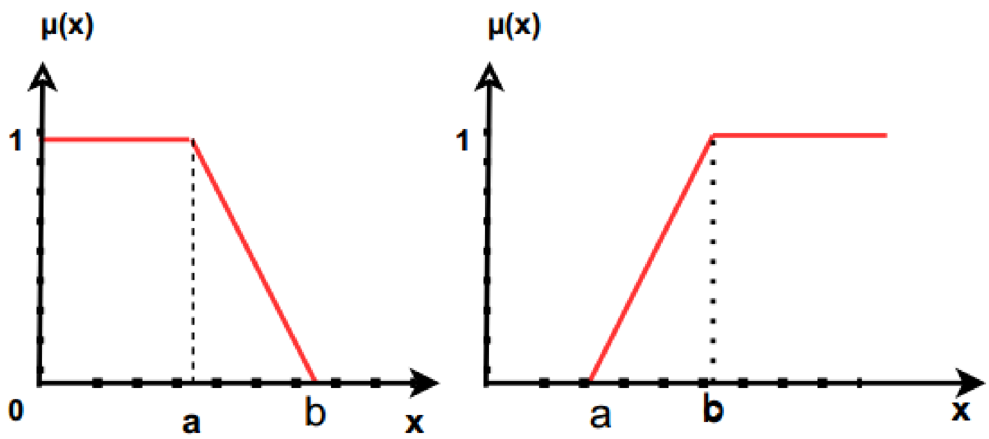

Figure 4.4: Left-shoulder and right-shoulder membership function.

4.2.2.3 Gaussian membership function

A Gaussian membership function is characterized by its central value c and its standard deviation (σ) [126]. The presentation and definition of the Gaussian membership function in Figure 4.5 and the following equation:

$$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right) \tag{4.5}$$

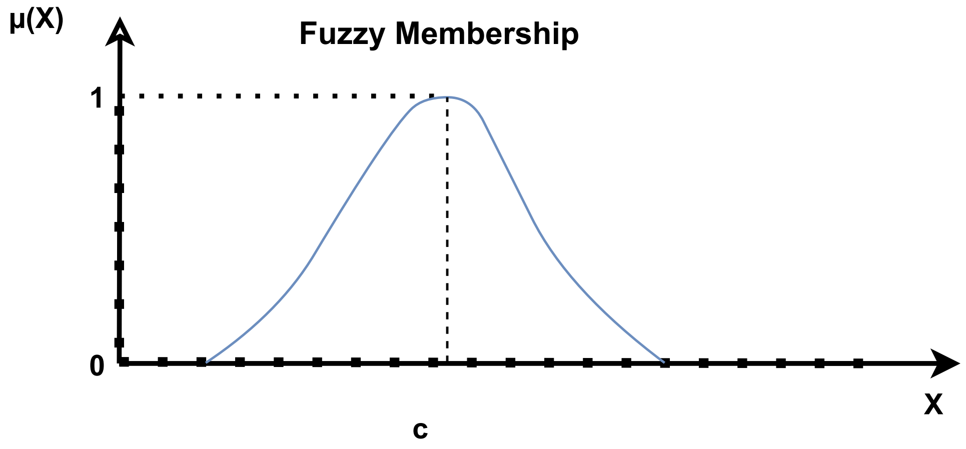

Figure 4.5: Gaussian membership function

4.2.3 Fuzzy Operators

The use of operators on fuzzy sets enables the determination of degrees of truth by writing logical combinations between various fuzzy concepts. These operations are expansions of the ones that have been suggested for the classical set. The operators who are identified as "from Zadeh" are the most widely used as follows:

- Intersection (AND): The minimum degree of truth for both X and Y is represented by the truth value of the statement "X AND Y".

$$\mu(XANDB) equals MIN(\mu(X), \mu(B)).$$

- Union (OR): The degree of truth of the statement "X OR Y" is equal to the maximum of the degrees of the truth of X and Y:

$$MAX((X, Y)) = (\mu(X) OR \mu(Y))$$

- Complement (NOT) The negation is the logical operator that corresponds to a set's complement.

$$\mu(NOTx) equals 1 - \mu x(x)$$

4.2.4 Fuzzy inference system

The principle concept of a fuzzy system is its ability to compute output parameters based on a set of rules formulated in natural language by providing the system with a set of rules formulated in natural language. The general architecture of the systems based on the fuzzy theory shows in Figure 4.6, which its implementation requires three essential steps [8].

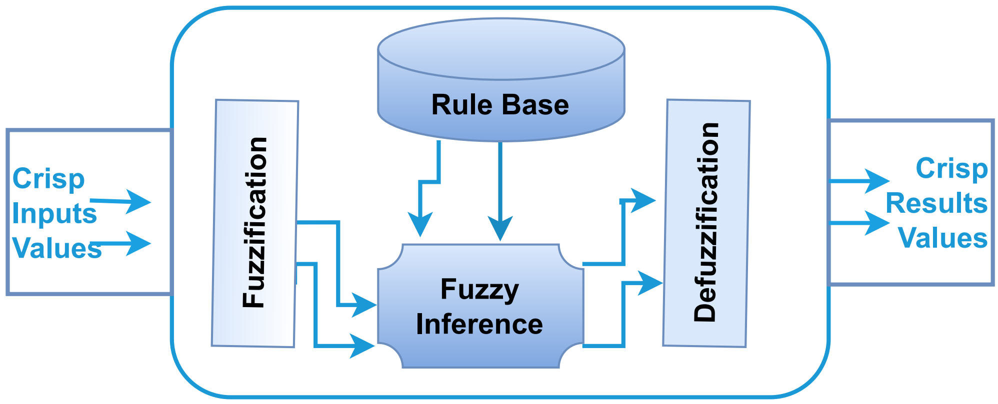

Figure 4.6: Fuzzy inference system [8]

4.2.4.1 Fuzzification

Fuzzification is a very important operation responsible for transforming each crisp input value into a fuzzy value. It represented by a degree of membership that determined by a corresponding membership function [129]. Fuzzification enables converting a numerical value(real domain) into a linguistic variable (fuzzy domain). To fuzzify, domain experts must include membership functions that define a numerical value's degree of membership to a linguistic variable.

4.2.4.2 Fuzzy Inference

The fuzzy inference engine is another name for the fuzzy inference process. It combines a set of if-then rules, membership functions, and reasoning mechanisms to derive the fuzzy output [8]. The fuzzy rule base, or the place where these rules are organized, consists of a set of linguistic statements explaining how the fuzzy inference system should recognize an input or control output. The fuzzy rule base is based on the knowledge extracted from domain experts. In the inference engine, all conceivable pairs of values connected by logical operators (e.g., AND, OR, NOT) are likely to be taken into account when organizing the rule base. Typically, rules in rule-based systems resemble the following: *If A1 and A2 an ..An then C1 and C2 and...Cm Ai* represent the rule's antecedents and *Cj* represent the rule's conclusions, Where the antecedents are the rule's presumptions that must be true to establish its conclusions. The two most popular fuzzy inference systems used in various fuzzy logic applications are:

4.2.4.2.1 Mamdani Mamdani-type fuzzy inference system [130] is one of the first fuzzy systems. Professor Ebrahim Mamdani of London University proposed it. Mamdani-type fuzzy inference system is widely utilized in decision support applications because the rule base has an interpretable and perceptive nature. A fuzzy inference system is more understandable, and its rule basis is simple to read. It can use Multiple-input, single-output, and multiple-input systems directly.

4.2.4.2.2 Takagi, Sugeno, and Kang engines Sugeno [131] proposed a fuzzy inference method that guarantees the continuity of the output. This inference method is very efficient in applications involving linear, optimization, and adaptive techniques. It is very similar to the Mamdani method but only altered one rule as a result (resultant). He employed a mathematical function of the input variable rather than a fuzzy set. It is only adequate to use with state multiple input-single-output systems. However, most applications reported in the literature use the fuzzy inferences defined by Mamdani, which have a simple structure and expressive power due to being more intuitive.

4.2.4.3 Defuzzification

Defuzzification permits production output values from fuzzy variables by using the rules located in the rule-base. The result is crisp values required in other systems processing or actor's interpreting. Defuzzification typically consists of two stages: the first involves merging the common linguistic variables using a fuzzy logic operator selected by the system developer. The maximality operator is frequently used with the OR logic operator. The same data are also characterized by several linguistic variables, each of which has a membership function. In the second stage, Defuzzifying data implies using a defuzzification technique to determine the best quantitative value in accordance with the membership functions of the linguistic variables. Among the most used methods is the Center of Gravity Method (COG) [132], which provides a numerical value based on the center of gravity of the fuzzy set. Other methods, such as the Average Maximum Method (MOM) and the Average Weighted Method (WAM), are valid for fuzzy sets with symmetric output membership functions.

4.3 Fuzzy ontology

Fuzzy ontology is an extension of classical(crisp or precise) ontology. Its foundation is the incorporation of fuzzy logic into developing the crisp ontology to represent and reason about imprecise data. Zongmin et al. define fuzzy ontology as *"a fuzzy ontology is a shared model of some domain Which is often conceived as a hierarchical data structure containing all concepts, properties, individuals, and their relationships in the domain, where these concepts, Properties and so on may be defined imprecisely"* [133]

Based on this definition, fuzzy ontology is built by adding to classical ontology components that conform and use concepts of fuzzy set theory, like the fuzzy concept, fuzzy object property, fuzzy datatype, and fuzzy axioms, to ensure the representation of the fuzzy parts of a universe of discourse.

4.3.1 Components of fuzzy ontologies

Precise concepts, fuzzy concepts, precise relations, fuzzy relations, axioms, and instances are all components of a fuzzy ontology. The precise concepts and relations are identical to those found in a classical ontology and serve the same functions.

4.3.1.1 Fuzzy concept

A fuzzy concept is an essential component of fuzzy ontology. It is considered a fuzzy set. It has an instance that belongs to it with a certain degree of membership [134]. In practice, this degree denotes the extent to which a given entity should be considered an instance of the concept. The degree is calculated according to the computational formulas related to the membership function defined for the concept.

For example as in Figure 4.7, *"YoungPatient"* is a concept that represents the instance of patients whose age is medium and is able to categorize them as Young, because "Young" is a vague predicate; the concept *"YoungPatient"* is similarly vague; therefore can be represented as a fuzzy concept, as in "Patient who has 40 years aged, it represents an instance of a *"YoungPatient"* with a membership degree of 0.8." [10]

4.3.1.2 Fuzzy relations

Similar to the reason for defining fuzzy concepts, the relations between concepts or individuals are not always clear. A fuzzy relation allows linking the instances of concepts according to a degree of strength that has a value between 0 and 1 to represent how this relationship should be considered. These degrees are also calculated using the formulas related to the membership function that is defined for this relation [134].

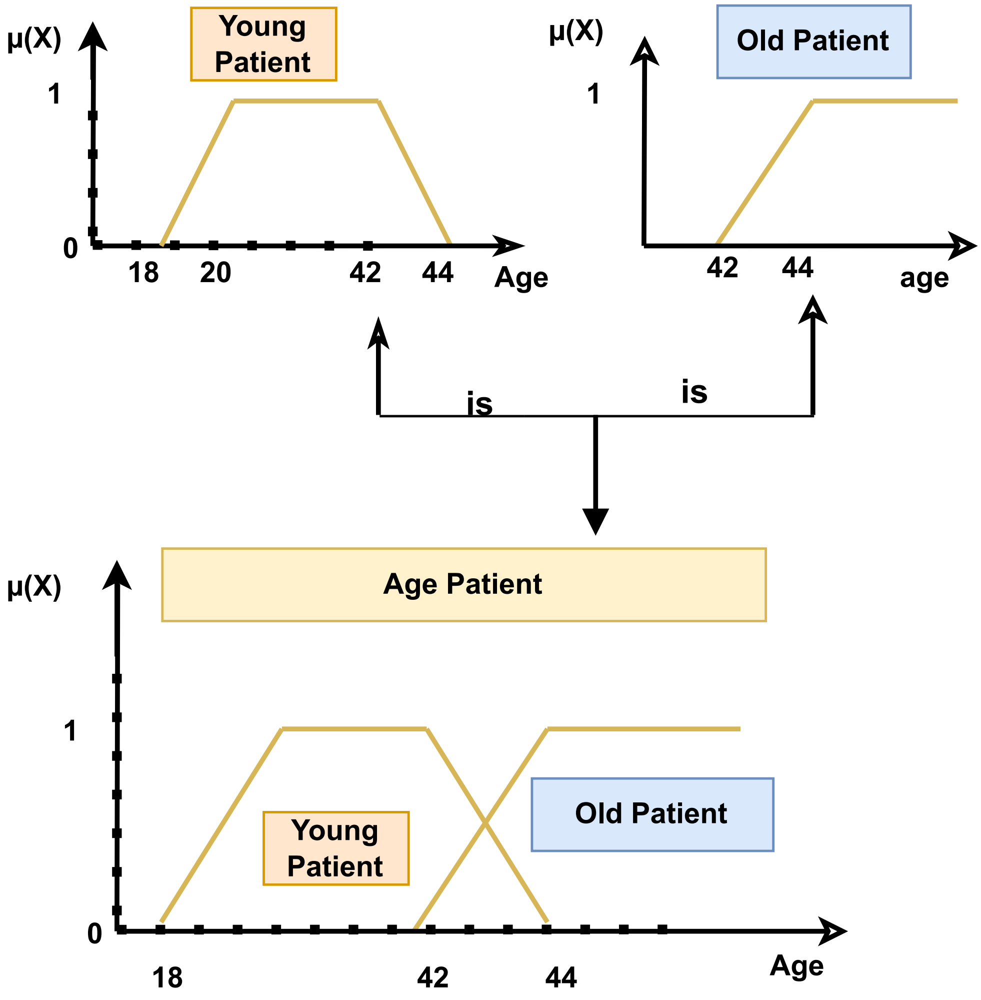

Figure 4.7: fuzzy Concepts [9]

In ontology formulation, There are two types of relations; the first is fuzzy object relationships. It permits linking the instances with a certain degree, as shown in Figure 4.8, "Patient1 ModerateDrink Drug1" with a certain degree of 0.6 where patient1 instance of a patient concept, and the second is the fuzzy data type relationships that its instances or individuals are assigned a literal value with some degree. Additionally, it links a class's instances to fuzzy data types.

4.3.1.3 Fuzzy data types

They specify data types that have vague meanings. It is represented by a set of vague terms that can be used in the ontology as attribute values. Fuzzy data types are also called fuzzy concrete domains or fuzzy linguistic variables. The necessity for developing

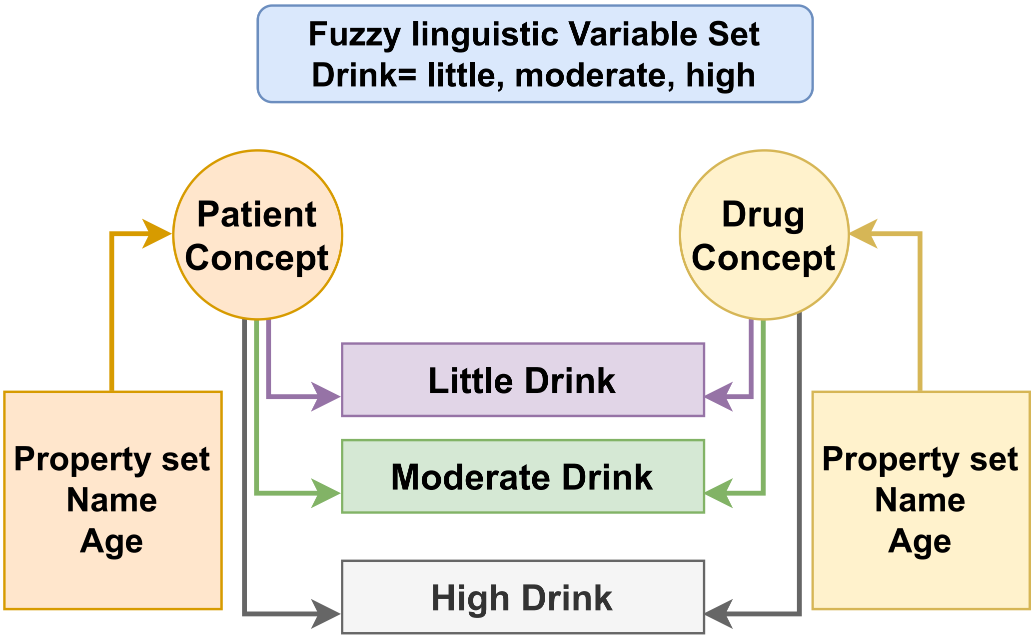

Figure 4.8: Fuzzy relationship example [9]

fuzzy data types is related to the availability of numerical values in ranges.

4.3.1.4 Fuzzy axiom

The fuzzy axiom is another component whose evaluation is linked to the degree of truth. It allows the expression of some states according to a certain degree in [0, 1] and differs from the state evaluation of classical axioms (true or false). The fuzzy axiom helps to represent the relations between concepts, relations, and individuals and helps to represent fuzzy ontology structure information. It also can offer more of these objectives than usually expected by allowing expressing formulas for determining membership functions.

4.3.1.5 Instance

An instance belonging to a fuzzy component is uncertain in fuzzy ontologies. It is decided by a membership function and a probabilistic variable with values between 0 and 1 [134]. For example, a patient has 66 years. We can categorize him as a "patient-old" with a degree of membership of 0.9 using the membership function.

4.4 Approach to the construction of fuzzy ontology components

In the literature, there are many approaches to designing fuzzy ontology. As fuzzy ontologies are an extension of classical ontologies, some of these construction techniques are based on this principle. They follow the same steps used for the design of classical ontologies that consist of specification, conceptualization, formalization, and implementation phases.

The methods allow the construction of an ontology from scratch, especially those developed for a specific domain. The most popular of them is the Fuzzy OntoMethodology [134]. This methodology is a method for designing and developing fuzzy ontologies. Indeed, this method allows defining crisp and fuzzy concept dictionaries from the corpus of fuzzy semantic data. Thus, this method assists the ontologist in constructing fuzzy ontologies. The methodology inspires the steps of this method for designing precise ontologies from methontology [5].

Another method is defined as a fuzzy ontology for a specific domain, such as AlzFuzzyOnto [135], which is specific to Alzheimer's disease (AD). The AlzFuzzyOnto ontology is based on the mind ontology for the construction process. The concepts of the mind ontology are used as a core domain ontology. Thus, the uncertainty data inherent in each concept and the relationship of the ontology are examined with the support of domain experts.

Developing ontologies from scratch requires effort, time, and cost. On the other hand, reusing all or part of them reduces this requirement and provides ontologies of good quality through improving continuous ontologies. Unfortunately, there are not many fuzzy ontologies in contrast to all classical ontologies that are defined. Converting these classical ontologies to fuzzy ontologies is represented an adequate solution to overcome this problem [10]. Following a methodology to convert these classical ontologies to fuzzy ontologies is necessary.

4.4.1 IKARUS-Onto methdology

IKARUS-Onto (Imprecise Knowledge Acquisition Representation and Use) [10] is a performing methodology for extending the classical ontology to a reusable and shareable fuzzy ontology. It allows to development of the ontology in the most appropriate way to ensure good quality in terms of accuracy, sharing, and reuse of the process results. The methodology is based on correctly identifying and modeling vague knowledge accurately using a fuzzy technique. It allows for the conversion of a classical ontology into a fuzzy ontology without changing the fundamental activities of this classical ontology. The methodology supports the ontology developer and the domain experts in recognizing and modeling the vagueness of the domain in order to make the content of fuzzy ontology reflect the vagueness as accurately as possible. The methodology focuses more on the development process and final content of the fuzzy ontology. Ontology developers can use it to pick the degree of membership values, recognize ambiguous ontology elements, and accurately identify fuzzy domain knowledgee. The methodology is reinforced by language and tools [136] to facilitate the development of fuzzy ontologies. For converting the fuzzy ontology, The IKARUS-onto provides a five successive concrete steps: acquiring crisp ontology, establishing the need for fuzziness, defining fuzzy ontology elements, formalizing fuzzy elements, and validating fuzzy ontology(see Figure 4.9).

Step 0 (Obtain classical ontology) : This step represents the reusing of all or part

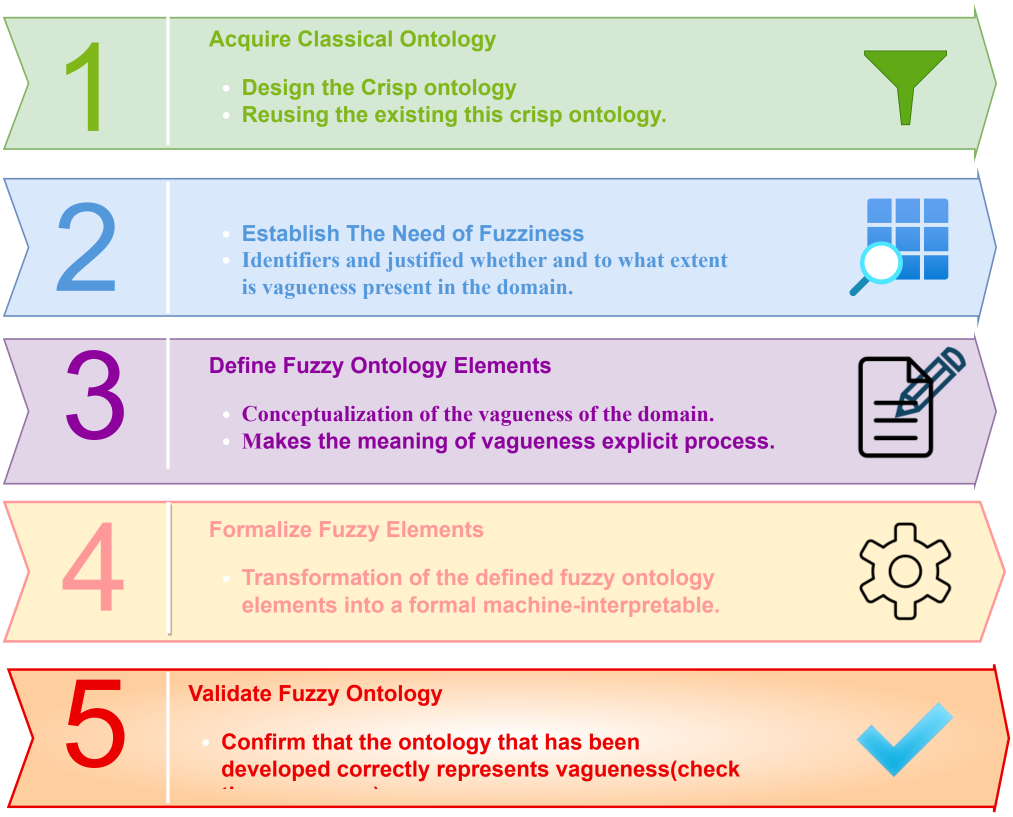

Figure 4.9: Ikarus-Onto methodology for designing a fuzzy ontology [10]

of classical ontology and may use the construction method for classical ontology in case of missing an existing ontology.

Step 1 (establishment of the need for fuzziness) : The need for fuzzification step

is necessary for an attempt to design fuzzy ontology. It consists of identifying whether and to what extent vagueness is present in the domain. It should be assisted by a domain expert to define the fuzzy part of the ontology. It consists of the Identification of vague concepts, Identification of vague relations and attributes, and Identification of vague attribute value terms. A concept or relation is vague means, if a concept or relation admits borderline cases in a given domain, that is, if there are individuals for whom it is unclear if they instantiate the concept or if there are individuals in pairs, it is unclear whether they belong to the relation. The same is true for vague attributes and individual pairs.
Step 2 (definition of fuzzy ontology elements) : For describing each kind of fuzzy ontology element, IKARUS-Onto suggests a specific process and description template. In order to make the identified fuzzy elements shared and reusable, this description aims to give them a clear and explicit vague meaning. The detailed identification of the domain's vague knowledge and its clear description using fuzzy ontology elements constitute this step. It involves the conceptualization of the vagueness of the domain and makes the meaning of vagueness explicit process through the identification and description of the vague knowledge of the domain, especially with the ontology's fuzzy relations and fuzzy attributes. In the following, the steps for defining fuzzy relations and attribute duties [10]:

- List all relationships and attributes that have ambiguous meanings.
- Determining the type of vagueness for each relation or characteristic. Determine the dimensions along which the element is imprecise and if it has any degree of vagueness.
- Outlining the precise significance of each element's uncertainty. The distinction between the dimensions may or may not be significant if the element has degree-vagueness over many dimensions. If so, each dimension would require its own unique fuzzy element to be defined.
- A definition of the anticipated meaning of each element's fuzziness degrees. The fuzzy degree of a pair of instances (or instances and literal values) associated with one another is a useful approximation of the amount to which the pair's value for the given dimension places it within the application boundaries of the elements, if the fuzziness is caused by degree imprecision.
- Give each pair of instances (or literal values) that create each element a specific fuzzy degree.

The procedure used to define fuzzy concepts is similar to that used to define fuzzy relations and characteristics, but there is a significant distinction. The vagueness of a concept is frequently "owed" to a previously specified, vagueness relation, feature, or word. If so, one of the concept's causal components can be used to directly deduce the definition of the concept's vagueness.

Step 3 (formalization of fuzzy ontology elements) : The transformation of the defined fuzzy ontology elements into a formal machine-interpretable form through some corresponding fuzzy ontology language. The range of fuzzy ontology items and the range of fuzzy reasoning capabilities are crucial aspects of a fuzzy ontology language that should be taken into consideration.

Step 4 (validation of fuzzy ontology): In order to ensure the ontology that has developed is represented the vagueness of the data in an adequate and correct way, the validation process is required step allows checking the following features:

- The ontology has fuzzy elements that convey a meaning which is indeed vague in the given domain(Correct).
- The ontology captures all the vagueness of the domain that is represented(Complete).
- The ontology does not include contentious information regarding the ambiguity of the domain (Consistent).
- The degrees of fuzzy elements must approach their ontology's imprecision in an intuitively accurate way. (Accurate).

4.5 Tools for Fuzzy Ontologies

Some tools have also been created for fuzzy ontologies. Here, a few examples are discussed:

4.5.1 KAON Project

It is expanded by Calegari and Ciucci [137] to include fuzziness in the ontology definition. It is based on a suitable Fuzzy Inspector Panel that represents a fuzzy entity, membership level, and the number of updates that are used to create a fuzzy inspector. This tool for development is based on their approach, which enables updating fuzzy numbers via query.

4.5.2 Fuzzy Protégé plugin

It was developed for Protégé 3.3.1 by Ghorbel and colleagues [9] as a fuzzy plug-in. The plug-in supports the instantiation of fuzzy concepts and roles and permits the definition of parameterized membership functions. Additionally, it enables automatic membership degree computation and fuzzy ontology querying based on fuzzy criteria.

4.5.3 Fuzzy OWL2 Protégé plugin

It was developed by Bobillo and Straccia [138] using OWL 2 annotation properties named fuzzyLabel to encode fuzzy ontologies. The plugin is GUI (graphic user interface) and compatible with Protégé versions 4.1 and 4.3. Additionally, they create a parser for converting OWL2 annotations that represent fuzzy information into a language understood by a number of reasoners, such as fuzzyDL [136] and DeLorean.

The reasoners that accompany this plugin and support reasoning in both fuzzy logic and fuzzy rough set enable us to check the consistency of the fuzzy ontology and also query it. FuzzyDL-based query was executed through the Fuzzy OWL2 tab to output its result for confirmation of the Consistency. The developers publish the plugin and the parsers for using in the Protégé on their website [139].

4.6 Conclusion

Health data generated by IoT devices and transferred for processing or representation can exhibit vagueness and imprecision in certain application domains. In this chapter, we have presented fuzzy logic as a solution that deals with this type of data. We have also presented the fuzzy notion of ontology and their component to facilitate the representation of data with vague and imprecise characteristics, as well as we have presented their fuzzy component concepts, fuzzy relationships, etc. We have also presented the methodology of construction of fuzzy ontology and the tools that help the development and visualization.

Although the introduction of fuzzy logic into the semantic web of things is a solution that will enable the processing of fuzzy and imprecise data, the framework, due to the increasing number of IoT devices, generates a large amount of RDF data that requires processing in terms of storage and retrieval.

Chapter 5

Exploring BIG Data and BIG RDF Data

5.1 Introduction

The advent of the World Wide Web in the early 1990s led to exponential data growth. This growth was further fueled by revolutionary developments in various technologies, including cloud computing, the proliferation of social media, and wireless communication technologies, which have enabled the collection of massive amounts of data. Data's exponential growth has created challenges in processing and analyzing the expanding data. The concept of Big Data emerged in a scientific paper presented at the Association for Computing Machinery (ACM) in 1997, highlighting the technological challenges of visualizing these large datasets. Since then, this emergence has highlighted the importance of developing effective solutions to enable the processing and analysis of such data. These first steps towards Big Data came from "search engines" such as Google and Yahoo, which operate as online information retrieval actors. After that, most companies began to use Big Data because of its growing benefits, like Amazon and Facebook quickly adopted their methods.

One of the features of Big Data is its heterogeneity, which can cause data integration and analysis issues. To address this, the semantic web represents and exchanges data in RDF (Resource Description Framework) format. Despite the adoption of this standardized representation format, which facilitates the combination and analysis of diverse data sets, the size of RDF Data has grown very quickly. This state poses challenges to the storage and processing of this massive data. In addition, there is a need for an efficient method of extracting information from Big RDF Data.

5.2 Big Data

5.2.1 Definition

In the literature, many definitions describe Big Data and share the same concepts. We cite the principal ones: Gartner provided the first definition of Big Data in 2001. He defines Big Data as *"Large, high velocity and/or high-quality information resources that require new forms of quality that require new forms of processing to improve decision-making, discovery, and optimization processes"* [140].

Oracle also defined Big data as *"larger, more complex data sets, especially from new data sources. These data sets are so voluminous that traditional data processing software just can't manage them. But these massive volumes of data can be used to address business problems you wouldn't have been able to tackle before"* [141].

Another definition provided by the national institute of standards and technology (NIST) defines Big data as *"extensive datasets – primarily in the characteristics of volume, variety, velocity, and/or variability that require a scalable architecture for efficient storage, manipulation, and analysis"* [142].

Generally, all these definitions address the mass of digital data generated by different sources, the best methods to manage and process them, the impact of this technology on different domains, and describe its characteristics.

5.2.2 Big data characteristic

The authors Zikopoulos and Eaton use the 3Vs model (Volume, Variety, and Velocity) to describe the Big data characteristics [143]. Furthermore, Lomotey described the 5Vs model (Volume, Variety, Velocity, Value, and Veracity), extending from the previous 3Vs model [144]. The Vs model is then extended each time, depending on new technologies used to understand the data and assess the outcomes. To understand these Vs depicted in Figure 5.1, we explain the characteristics of Big Data in the following.

5.2.2.1 Volume

Refers to an enormous volume of information gathered and generated from infinite sources. The volumes of big data require separate data storage and processing, as well as additional data management and preparation processes. The increased capacity of storage represents the growth of Big data volumes. According to Statista, storage capacity has grown from 6.7 zettabytes in 2020 to 8 zettabytes and is predicted to reach 16 zettabytes in 2025 [145].

5.2.2.2 Velocity

The data generated from the billions of connected sources is always changing and evolving. It creates a problem for retrieval, processing, and exchanging it in real-time using traditional devices and systems. Velocity refers to the speed of Big data, which is generated, collected, and processing in real-time to provide immediate responses. The speed of data generation and processing increased, and real-time processing improved application efficiency, providing immediate responses and maximizing the value of the information.

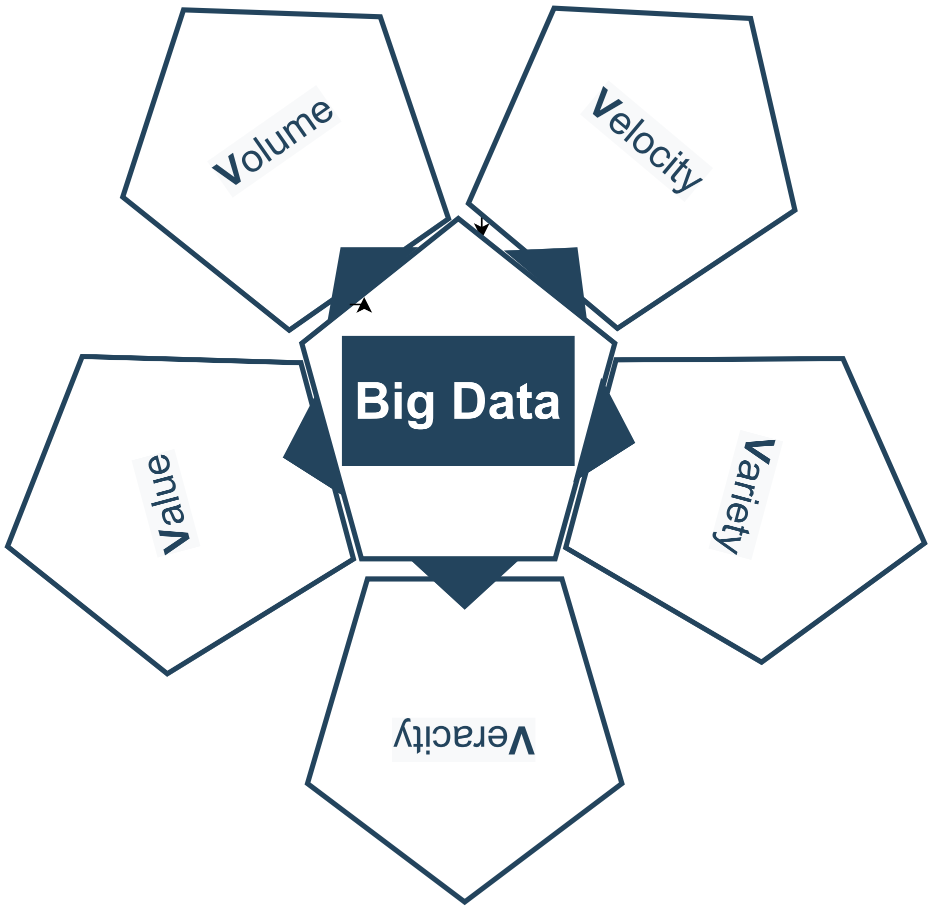

Figure 5.1: Big Data characteristic

5.2.2.3 Variety

The different sources are generally heterogeneous and generate different types of formats. These formats of data are classified into three groups. The first is called the structured format when data originate from relational databases. The second is semi-structured when data come from online logs, emails, social media feeds, etc. Finally, the unstructured format is when data have types of photos, audio, video, sensors, etc. Unfortunately, The unstructured format of data is the most generated compared to structured data, which has led to the development of new approaches capable of analyzing it quickly.

5.2.2.4 Value

Oracle introduced this feature as a fourth dimension to the Big Data 3Vs. Big Data architectures are designed to cost-effectively extract relevant data from large volumes of a wide variety of data, allowing high speeds to capture, discover, and or analyze. As a result, the value provided by Big Data has allowed companies to increase revenues, reduce operating costs, and better serve customers.

5.2.2.5 Veracity

Most data collected characterized its incompleteness, imprecision, delay, inconsistency, and subjectivity that, caused uncertainty and unreliability. Two companies, IBM and Microsoft, offer it as another criterion for assessing the dependability of data from various sources. Veracity refers to a leader's level of trust in the data that has been processed and studied to make a decision; in other words, it is the data's conformity to the truth, accuracy, and certainty.

5.3 Big Data application cases

Big Data is used in many domains, such as industry, finance, transportation, banking, insurance, and telecom. Some of these examples are listed below:

5.3.1 Transportation

Big data can help the transport domain in its applications. It allows the use of various types of source data to smooth out traffic and avoid congestion. It improves the safety of passengers and provides them with suitable travel planning. It also allows the real-time visualization of transportation data, including buses, cars, trains, and planes. It can play a role in enabling intelligent transportation systems (ITS), including autonomous vehicles, cooperative vehicles, satellite positioning systems, etc [146].

5.3.2 Health care

Big data provide healthcare domain with many advantages that can improve the health care quality, for example:

- Exploiting data stored for years allows an understanding of the causes of disease and its effects.
- Follow-up of patients (patient medical records), especially patients with chronic diseases, to predict the high risks that may befall them.
- Predicting and detecting diseases earlier than they occur through stored medical information.
- Big data can provide valuable insights and information for health actors such as doctors and policymakers to make informed decisions to improve health services.
- Big data can contribute to drug development by accelerating drug discovery and development by identifying potential drug targets.

5.4 Internet of things and Big Data

Big Data and the Internet of Things are two closely related advanced technologies. According to International Data Corporation (IDC), IoT will generate 79.4ZB of data in 2025 [147], which is expected to increase in the succeeding years. Moreover, this data must be read, exploited, and transmitted on time. To exploit a massive amount of data, using Big Data is necessary. This technology allows real-time analysis of the data generated by IoTs and therefore optimizes the use of this technology. It also enables IoT developers to raise the effectiveness of data processing and storage. On the other hand, the Internet of Things has added significant value to Big Data. As IoT technologies significantly impact our daily life and improve and ease it, developers will demand more capacity in terms of Big Data, and this technology will become more critical. The Internet of things contributes to increasing the business value of Big Data, as stated by the International Data Corporation (IDC) [148] *"Global Spending on Big Data and Analytics Solutions Will Reach $215.7 Billion in 2021"*.

The interaction between the two technologies can be seen in various disciplines, including health, where one of the main challenges of the health Internet of Things is to provide to improve decision-making. Integration aims to develop remote diagnosis and care systems that accurately diagnose diseases at an appropriate time from any place. The integration benefits from the Big data's efficiency in analyzing and extracting relevant data gathered from IoT sensors on a wide range of physical items to better understand diseases.

Although the internet of things has become a crucial source of Big data through the rise in the number of IoT devices employed in many fields, the variety of sensors generates a high variability of data types and formats due to a large of unstructured data that needs to be processed and stored. Data heterogeneity represents a significant challenge in both domains [149]. In Big Data, data coming from various sources and which may include organized, semi-structured, or unstructured data, is defined by the characteristic of the variety. Using the semantic web is one method for making sense of this variety of data in big data.

5.5 Big data and semantic web

The semantic web is a technology solution that can help overcome some challenges associated with managing and analyzing the vast amounts of data generated by IoT devices. The Semantic Web allows for the annotation of data with semantic metadata, which enriches the data with additional context and meaning. This metadata can be used to facilitate understanding and reuse of the data in a consistent and interoperable manner [150]. The semantic web is based on the ontology for knowledge representation and exchange. The ontology has been extensively used during the last 20 years in big data applications in a variety of fields [151], as depicted in Figure 5.2. in addition to the ontology, The semantic web is also based on other semantic technologies. Resource description format(RDF) is a standard and flexible model that enables the representation of the data[see section 3.3.2.3.] It has a simple structure based on triples of data (subject, predicate, and object), and allows data exchange among different systems. Although exchanging IoT data using the RDF format is rapid and easy, the size of RDF datasets is growing quickly due to the creation of Big RDF Data.

5.5.1 Big RDF data

Big RDF data is massive data consisting of data in RDF format. It is becoming increasingly popular because the RDF data format is known for its simplicity and adaptability, making it well-suited for describing and integrating various types of data from various sources. Consequently, many RDF datasets are produced and cover several domains,

Figure 5.2: Using of ontologies in big data applications [11]

and each dataset contains billions of RDF triples addressed to describe the applications contexts, for example:

5.5.1.1 The DBpedia

It is a huge dataset that, as of 2014, had 3 billion RDF triples. It includes a wide range of topics, such as people, music, places, movies, organizations, etc [152]. It includes structured, multilingual information that was taken from Wikipedia, which accounts for 14% (or 1 billion triples) of the whole DBpedia. Numerous ontologies are included in DBpedia, which also offers a collection of user queries using SPARQL that are chosen from the most popular queries.

5.5.1.2 Lehigh University Benchmark (LUBM)

It is a large RDF dataset designed to reflect the academic world. It is made up of a sizable amount of repeating, customizable synthetic data that is distributed over several files [153]. Using an artificial data generator, it consists of synthetic data for universities, their faculties, their professors, their student, their courses, etc. The dataset is designed to make it easier to evaluate Semantic Web repositories uniformly and organized manner using various reasoning techniques and storage methods. Based on the university domain ontology, the dataset is used to evaluate the effectiveness of RDF storage by offering test queries.

5.5.1.3 Linked Sensor Dataset

RDF Datasets for sensors and sensor observations were developed at Kno.e.sis Center and transformed from weather data at Mesowest. It includes expressive descriptions of 160 million observations from 20,000 weather stations in the United States [154]. The dataset is divided into many subsets that represent weather information for specific hurricanes or blizzards that occurred in the past. The dataset has over 1.7 billion RDF triples and uses the SSN ontology to represent sensor and observation data.

5.6 RDF data management systems

RDF is based on the triple format (s, p, o), which can be organized into a table with three columns: subject, predicate, and object. By taking advantage of the 40 years of research expertise in the database community, it is possible to store RDF data in an RDBMS (relational database management system) by adopting a particular schema designed for RDF data. To store RDF data, researchers have proposed a set of approaches; they begin by proposing to store all RDF data in a single table with three columns [155]. To overcome the self-joins challenge, they combine all properties of similar objects in the n-ary table. The latter approach occurs its weakness in dealing with multi-valued properties and potentially many nulls value [156]. Another approach called a "vertically partition table" (binary table) provides for each unique property by a two-column table to avoid multi-value and null value problems [157]. It also used the ID for more efficiency although the expensive insert problem. To fast join the data stored and enable efficient processing, some researchers have proposed creating an index for each possible combination [158]. The approach needs five times more storage and has a weak performance where it is necessary to accede to the storage disk. Based on this approach, and intending to store RDF data and manage the queries efficiently, many RDF data management systems are currently available. An example of the most popular of these triple-store solutions available are:

5.6.1 Jena

It is an open-source Java framework for creating Semantic Web and supporting the linked data applications and support for inference (using RDFS, OWL-Lite, or custom rules). The framework has a component used for querying and storing RDF called Jena TDB. The component provides a high-performance RDF store and comprehensive SPARQL enables converting with SQL DBMS (SDB) [159]. It integrates with an open-source SPARQL server built inside the Apache Jena framework called Apache Jena-Fuseki. The latter may function as a system service and deliver queries and modify datasets by accessing the endpoint on the Web.

5.6.2 GraphDB

GraphDB is a W3C-compliant semantic graph database that is a useful tool for commercial applications. Using RDF triplestores, the solution places a significant emphasis on the ability to quickly model data, aggregate data from many sources, study the relationships between data points, and deliver data to many businesses. Companies can use GraphDB to quickly build scalable knowledge management systems, recommendation engines, and other data-intensive applications that integrate and explore data [160].

5.6.3 Blazegraph

It is an open-source platform also known as Big Data and has a commercial license. It provides efficient triple storage of RDF data [161]. The platform, which was wholly written in Java, supports the query, update, basic federated query, and service description RDF/SPARQL 1.1 family of standards. From small applications with integrated storage to big standalone applications, the Blazegraph solution meets all application demands. It is widely utilized in both commercial and government applications to enable cyber analytics. The platform is under development and usage for Fortune 500 clients, including EMC, Autodesk, and several more companies.

5.6.4 RDFLib

A pure Python library works with RDF data. It enables parsing, serializing, storing, and manipulating RDF data [162]. Many companies use it to analyze and store data on the semantic web. It also implements both queries and updates using SPARQL 1.1. RDF data management systems use SPARQL queries to retrieve RDF data. The retrieving process requires passing a three-step. The first step consists of converting the SPARQL query to an SQL query, and the second step provides the SQL query with an answer from the RDBMS. The third step is the inverse of step 1, when the SQL results are transformed into the SPARQL result generated.

5.7 Indexing

The semantic web of things framework enables the processing of data captured by IoT devices to leverage heterogeneity and ensure interoperability. With a large number of these devices, a huge amount of data has been processed by the SWoT framework. Searching without using organization structure in the output of the SWoT framework means that users need to browse all this data to find the information desired, which can take a lot of time and effort. Indexing is a method of structuring data that allows locating information quickly that corresponds to a specific request. In the literature, a wide variety of indexing techniques can be found [12]. Tree-based methods have emerged as improving the speed and accuracy of IoT data processing, leading to better decisions. Tree-based indexing involves building a binary or multilevel tree where each node represents a subset of the data. The data is recursively partitioned into subsets until each node contains a small amount of data that can be searched quickly and easily. Many trees kind are used in tree-based approaches, which aim to organize structured data. Among this approach's most commonly used trees are the R-tree, and B-tree families.

5.7.1 R-tree

The R-tree is a multi-level tree structure used for geographic database organization and search [163]. It handles spatial queries, such as those for points, lines, or polygons in a specific geometric space, very effectively. To eliminate dead zones, the R-tree depends on recursively dividing the data into numerous minimal bounding rectangles (MBRs). Except for the leaves, which may include a range of geometric objects, each node has a fixed number of regions. When a geometric object is inserted into the tree, it is placed in the region that contains it and the tree structure is updated to reflect this change. The R-tree includes high efficiency in spatial data retrieval, optimized memory usage, and good scalability. It also effectively handles proximity queries, that is, queries to find geometric objects that are some distance apart. Unfortunately, because many MBR partitions overlap the performance of tree also negatively affected [164]. The tree can face drawbacks such as increasing the complexity of updating the tree when the number of objects inserted in the tree is large [164]. The tree is developed through defining its extension as in Figure 5.3 to improve its performance in such status or support specific data.

5.7.2 B-tree

The B-tree is a well-balanced tree structure for organizing and searching large amounts of data [165]. It is frequently used in file systems and databases to organize huge amounts of data. B-trees are very effective for large-scale data searches because of their low height, as searches can be performed quickly without having to go through many levels of the tree. To construct a B-tree, the sorted data must be recursively subdivided into blocks. Each block is stored in a node of the tree, with a pointer to the neighboring blocks. The nodes of the tree are organized hierarchically, with the blocks containing the fewest keys at the lowest level of the tree, and the blocks containing the most keys at the highest level. Nevertheless, the tree uses a lot of computing resources, requires a lot of storage space, and requires expensive maintenance. Many extensions have been added to the tree, each aimed at improving its performance for specific applications (see Figure 5.3).

The B-tree is a generalization of a binary tree. A binary tree is a B-tree in which each node has just the left and right children as its maximum number of offspring. A binary tree support organizing and storing sorted data, like numbers or strings. However, if the data distribution is not uniform, binary trees may become imbalanced and ineffective.

As the B-tree, the topology of the binary tree can be improved to give a more effective search and data insertion than with a basic binary tree to overcome these difficulties.

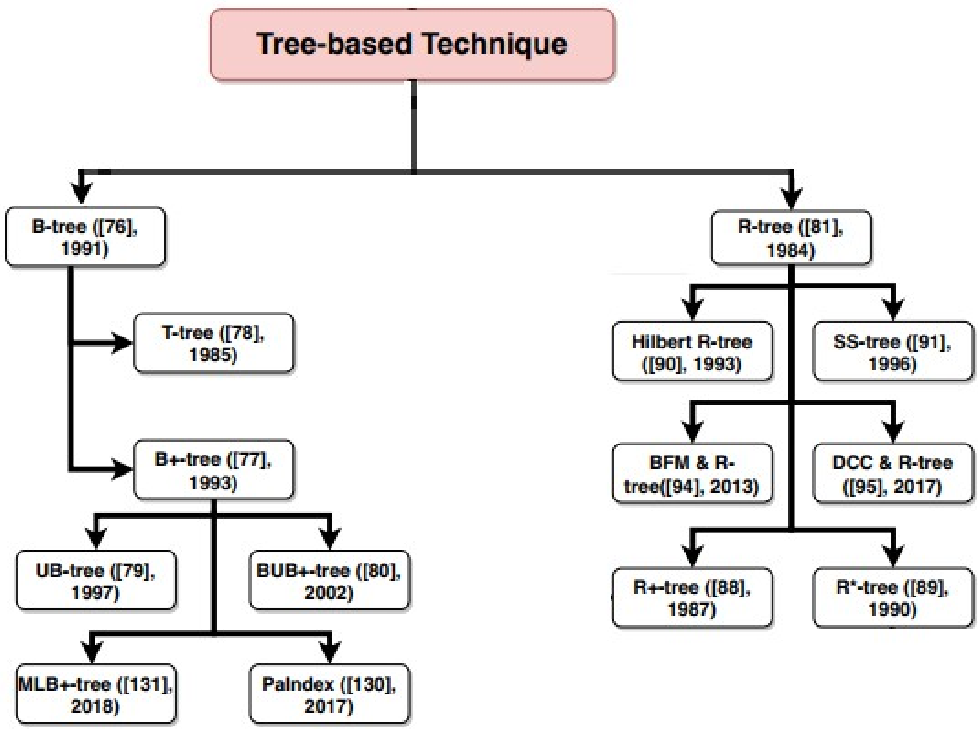

Figure 5.3: Taxonomy of the B-tree and the R-tree for data indexing ([12])

5.8 Conclusion

The emergence of Big Data in various fields of the industry highlights its importance, prompting many companies to exploit its capabilities to take advantage of the opportunities offered by this new technology. This chapter provides an overview of Big Data, its characteristics and its application areas. It also highlights the interaction between Big Data and the Internet of Things, where IoT technologies generate huge amounts of data, which requires Big Data in terms of storage capacity. It discusses the challenges of heterogeneity in IoT and Big Data, as well as the concepts of Big RDF data and relational database management syste(RDBMS). Finally, the chapter introduces an indexing approach that structures and organizes IoT data to facilitate quick search.

Chapter 6

Related work

6.1 Semantic web for handling heterogeneity challenge in IoT domain

The internet of things has led to a great revolution across various domains in recent years. It enables the development and improvement of these domains by providing them with capabilities that enable them to offer various services at adequate times. It is based on a set of tools that allows the collection, processing, and exchanging of data via the internet that allows overcoming the traditional barriers related to time, place and things, etc. To achieve more benefits from IoT marketing, many vendors produce owner devices, define various communications technologies, use their architecture, and develop their platforms and systems. This state creates an IoT fragmentation where users must follow the same vendor to benefit from IoT technology. According to Moriandi et al. [166] fragmentation might obstruct IoT technologies' successful development and integration. This phenomenon is called the "vertical silos" where these heterogeneous devices generate heterogeneous data characterized by their great difficulty in processing and exchange between different systems and domains due to lack of interoperability. In order to ensure interoperability amongst IoT systems and eliminate heterogeneity and ambiguity in the enormous volumes of data created by linked items, Barnaghi et al. [167] emphasized the relevance of defining and presenting IoT semantics. Soursos et al. [168] also discuss the necessity for cross-domain applications, IoT data management, and semantic interoperability to deliver helpful information and guarantee interoperability across IoT applications and reasoning. Semantic interoperability is a crucial interoperability level concern to the ability of two or more IoT systems (or components) to share data in an agreed format and in a meaningful way, on and off the Web [169]. It allows the system to understand the parsed message.

Achieving semantic interoperability leads to the emergence of the Semantic Web of

Things (SWoT) [110]. The SWoT integrates the semantic web into the internet of things to benefit from semantic web technologies assistance. With semantic web technologies, the IoT domain facilitates the interconnection of data and the acquisition of new knowledge to create intelligent applications. It also enables maintaining interoperability at the data processing, management, and storage levels.

Applying the semantic web of things creates a new problem in organizing and comprehending such diverse data within a constrained time in a scalable manner. This problem requires supporting the SWoT applications with a sizable memory for storage, a robust processing engine, and a wide bandwidth to manage and send this enormous amount of data. On the other hand, IoT devices' limited memory and processing power proved their inability to process the data acquired at the appropriate time. As a result, the semantic web of things needs technologies to provide these capabilities. Using the newest emerging technologies (cloud computing, edge computing, and fog computing) in SWoT infrastructure can create a semantic IoT system able to compute and store massive generated data.

Many works in the literature use cloud computing, fog computing, and Edge computing separately to enable more than processing and storage capabilities. They use a cloud for facility semantic web tasks. Poslad et al. [170] design an Early Warning System (EWS) architecture that simplifies the semantic functionality using a Semantic Cloud Computing approach. The proposed architecture facilitates the transmission of sensor-generated content to the cloud, where it is processed and enriched to generate meaningful high-level events. Instead of sending this data to distant cloud nodes or acceding to its ontologies and rule sets for reasoning tasks, other works aim to improve semantic computing in Fog nodes close to IoT devices. With this perspective, Chien et al. [171] propose using "Fog Computing" to divide the semantic reasoning process among a hierarchy of distributed fog nodes. They contrast the suggested distributed fog architecture with a centralized cloud system based on the use case of elderly care to confirm the efficacy of techniques. Petrovic et al. [172] propose using SMADA-Fog, a Semantic Model-driven Approach to Deployment and Adaptivity of container-based applications in fog computing. The proposed approach enhances application performance while enabling automatic code generation for managing fog computing infrastructure.

Other works use edge computing technologies to interact, understand, manage, and briefly exchange data semantically to benefit from its features like low latency, which lowers the network workload, scalability, and others. Sahlmann et al. [173] suggest using an extension of the oneM2M Base ontology for the IoT device descriptions to enable autonomous network management, service discovery, and aggregation. They introduce virtual IoT devices to hide real devices and their semantic framework deployed at the edge computing close to the network. Dhananjay Singh et al. [174] propose a semantic

Edge-based network model that is crucial for exchanging tactical and non-tactical pieces of information over the network. The proposed architecture aims to provide a secured zone to monitor soldiers' health and weapons conditions. In terms of processing, storing, and sharing information, providing a mechanism to handle the substantial amount of health data lead to using a wearable device as Edge computing. In order to facilitate wireless on-the-go communication across various wearable health monitoring devices, Nidhi et al. [175]offer SemBox as Semantic interoperability in a Box. It may establish a wireless connection with the health monitoring devices and receive data packets from them. Other works address the accommodation of semantic web-to-edge features. Konstantin et al. [176] to reduce the complexity of ontology, they provide a novel method for using ontology reasoning mechanisms on extremely resource-constrained Edge devices rather than in the cloud or fog. However, the loss of knowledge's immense value may result from the reduction in the number of concepts.

Creating a distributed architecture combining these various technologies allows for a large-scale IoT system, Seydoux et al. in [121] suggest applying fog-computing and cloud-computing to ensure scalable IoT data processing with SWoT. They propose a distributed method for rule-based reasoning that takes into account rules that adhere to the S-guiding LOR's principles. Su et al. [177] study how to distribute semantic representations and reasoning techniques of reasoning between cloud and edge devices to infer the activities of cars. The authors have proved that collaborative edge computing improved the system's performance. However, deploying this architecture requires an infrastructure that can sometimes be costly, and it must take into account the management of resources in such a way that the provision of appropriate service is consistent with the appropriate IoT resources.

6.1.1 Standard IoT ontology

Ontology is the cornerstone of the semantic web. It provides a formal and flexible model suitable for representing data captured by data sensors, and gives rich semantic descriptions that can be used for querying and reasoning [178]. Many works use semantic web techniques in the IoT domain. They focus on achieving semantic interoperability by defining an ontology. A part of existing ontologies is defined for addressing the appliance of IoT in a specific domain [179]. Another part that is defined to address the IoT domain is called "standard ontologies". Standard ontologies are a set of ontologies that represent the IoT domain by defining its component, tasks, communications, services, and other information. Many standardized IoT ontologies are currently used to represent sensors and their data in this context such as the SSN/SOSA [118], SAREF [124], and others [123].

Ontologies	Core concepts						
	Sensor	Actuator	Device	Service	location	Time	Unit of Measurements
Avancha 2004	*	-	-	-	*	-	-
Onto -Sensor2009	*	-	-	-	*	-	-
CSIRO 2009	*	-	*	-	*	-	-
SSN 2012	*	-	*	-	-	-	-
WSSN 2013		-	*	-	-	-	_
M3 2014	*	*	*	-	-	*	*
IoT-Lite 2015	*	-	*	*	*	*	*
IoT-O 2015	*	*	*	*	-	-	-
SAREF 2015	*	*	*	*	*	*	*
oneM2M 2015	-	-	*	*	-	-	_
FIESTA 2016	*	*	*	*	*	*	*
SOSA 2018	*	*	*	*	*	*	*

Table 6.1: Comparison the IoT ontologies using core concepts(*=Yes, -=No)

Table 6.1 and Table 6.2 list the ontologies that are primarily used in IoT applications (standard). The ontologies are listed with a year of publication to know their evolution and compared using some essential criteria:

As the concept is an essential part of ontology construction, Table 6.1 criteria are associated with "core concepts". These core concepts represent the typical sources, actions, communication, surrounding, etc., which play a crucial role in the IoT real world. Bajaj et al. [180] proposed and identify some of them, like sensors, actuators, devices, etc. The concept "Sensor" is used to represent sensing and sensor properties. The concept "Actuator" is also identified for representing actuating and actuator properties. We use the concept "Device" to describe the device, including sensors, actuators, or incorporating the two. We identify some essential other concepts like "Services" that represent how the IoT responds to users' needs, "Location" (referencing sensor location), "Time" (concerning the Time of actions), and "Units-of-measurements" concept to refer to the unit of measurement.

Table 6.2 comprises a collection of several crucial criteria about the construction of ontology, the domain it represents, and its adaptability to evolution. Some of these criteria are considered to be more crucial than others.

Standard: criteria enables the determination of the standardization of ontologies, where many existing ontologies specify their domain.

Heavyweight and lightweight: two criteria used to identify resource consumption as memory and processing. In contrast to Heavyweight, which represents most of the ontologies, lightweight refers to an ontology that describes important IoT concepts, which does not need computation and memory capabilities and can be used in an available low resource of the internet of things.
Complexity: this criteria is represented using its category (C: complicated, O: ordinary, S: simple). It refers to the difficulty of analyzing and transforming the data into useful information.
Support-alignment: criteria represents the ontology's ability to support mapping to one or more ontologies to allow data import and exchange.
Modularity: criteria presents the possibility of reuse and development of a part of the ontology and the ability to communicate with others.
Extension-from: criteria confirms the interconnection of ontologies that contribute to the ontology's development. The other criteria provide an observation about these ontologies related to the application domain, the ontology's use, and its characteristics and capabilities.

6.1.2 Synthesis

Table 6.1 and Table 6.2 show a qualitative comparison of the ontologies proposed to achieve the semantic interoperability of the internet of things. The ontologies detailed section(3.6.2) and analyzed via important criteria allow some conclusions can be inferred:

- Researchers have proposed several ontologies as a solution to enable semantic interoperability in the IoT domain since 2004. Subsequently, they continued to provide more efforts to improve the ontologies by defining new concepts to represent the different aspects of the IoT domain. The successive versions of the published ontologies are also the results of these efforts. However, these solutions still need further handling of other features to ensure their development and improvement.

- In the semantic web of things, we have more than 400 ontologies [181], most describing specific domains. This number and diversity of ontologies create a problem related to the communication and exchange of the data among them. It also prevents ontology developers from reusing or improving them. To avoid this problem, it should be defined a new ontology to represent the IoT domain using a high abstraction of concepts without due to the domain's lost essential features.

Ontologies	Standard	Heavyweight	Lightweight	Complexity	Alignment	Modularity	Extension	Reusable	Application, Purpose
Avancha 2004 [111]	-	*	-	C	-	-	-	-	- General domain - Define the condition of the sensor
Onto Sensor 2009 [112]	-	*	-	C	-	-	-	-	-General domain -Descriptive specification model for sensors
CSIRO 2009 [113]	-	*	-	S	-	-	-	-	- General domain -Describe sensors -Generic.
SSN 2012 [114]	*	*	-	C	-	*	-	*	- General domain - Using For observation -Define sensor propertie
WSSN 2013 [116]	-	*	-	O	-	*	SSN	*	-General domain -Present observation
M3 2014 [122].	-	*	-	C	*	-	SSN	-	-Cross-domain. -Extended to M3 lite.
IoT-Lite 2015 [117].	-	-	*	S	-	*	SSN	*	-General domain -Provisional description -Discovery of sensor
IoT-O 2015 [121]	*	*	-	O	*	*	SAN SSN	*	-Robot domain . . . etc. -General Actuation, observation -Energy consumption
SAREF 2015 [124]	*	*	*	O	*	*	-	*	-Energy domain, -smart building. . . -Describe devices, Extend to such ontologies
OneM2M 2015 [123]	*	*	-	C	*	-	-	-	-General domain -For services only, -Generic interworking.
FIESTA 2016 [120]	*	*	-	C	*	-	M3, SSN,..	-	-Unified ontology -Inadequate to cross-domain
SOSA 2018 [118]	*	-	*	c	*	*	SSN	*	-General domain -Generic model

Table 6.2: Comparison of IoT ontologies(*= Yes, - = No).

- The standardization of ontologies is one of the crucial criteria. It helps share the meaning between different domains and overcome the unambiguous concept. More recent works (between 2015 and 2019) tend to define a standard ontology with good quality as a reference to avoid the problem of interoperability of ontologies.

- The weight and the complexity of ontology are the dominant features of existing ontologies. They require high processing, massive storage memory, and more effort to formulate ontologies adapted to the constraints of the environment; for this reason, most ontologies frequently run in the cloud and get difficult with fog or edge paradigms.

- Recent works [117] [119] propose a light ontology to reduce complexity, optimize resources, and save time without missing the meaning of concepts. Nevertheless, these ontologies should overcome the problem of losing the richness of data.

- More than 60% of the ontologies in the table consist of an independent submodule called Modular. Modularization allows updating and reusing the parts of the ontology following the particular requirements and improving some parts without the need for developing a new ontology. However, the ontology needs a central module that functions as a system's core and methodology to join other modules.

- An ontology alignment is a set of correspondences between two or more ontologies. The standard ontologies SAREF [124], OneM2M [123], SOSA [118], and IoT-O [121] supported ontology alignment to facilitate the exchange of information for enhancing semantic sharing and encouraging the reusing of ontologies. But focusing only on the alignment of ontology leads to a limit with the update of ontology.

- Due to the importance of the extension of ontologies, 50% of ontologies in two tables are extended from SSN ontology, which presents a high level of abstraction that allows the description of devices, data, and services. The SOSA ontology shows an excellent example of the extension that extends from SSN and contributes to the development of other ontologies.

- The complexity and diversification of ontology are related to the concepts. At the same time, the effectiveness and the comprehensive ontology are up explicitly to the concepts used. For that, the authors in works [124] [123] [114] deal with this challenge by defining the core concepts required for developing an IoT application and extending them according to the advancement of the IoT domain and changing service requirements.

- The reuse of one of these ontologies is an appropriate solution to enable the semantic web of things, and specifically, the SAREF ontology is the most suitable to respond to the previous criteria.

6.1.3 IoT ontologies for representing health care and public transportation domains

As the IoT domain enables to integrate with other domains, several ontologies exist defined to represent the appliance of the internet of things in a specific domain. Health care is an essential domain that is related to people's lives. It provides them with the necessary treatment, prevention, and management to improve their physical or mental health. The incorporation of IoT in this domain is presented significantly in the health monitoring of patients, where many aging people and patients with chronic diseases need continuous monitoring [182].

By providing a health IoT ontology for patient monitoring, Rhayem et al. [183]provided an example of this work by highlighting a semantic representation of both connected medical objects and their data. The authors help decision-makers by analyzing the obtained vital signs and semantic rule reasoning. In order to facilitate interoperability and integration of health data, Reda et al. presented an ontology in [184] that represents health data from diverse IoT sources and permits autonomous reasoning by inference engines. To lower the development cost and benefit from the quality of the standard ontologies that are defined, Moreira et al. [179] employed a standard SAREF ontology to design its extension, known as SAREF4health. It allows IoT systems dealing with frequency-based time series to be semantically interoperable. The authors represented a time series of real-time ECG sensor readings to identify accidents involving truck drivers and demonstrate their ontology's effectiveness. The integration of IoT technologies to enhance the public transportation sector using ontology has been the subject of other research. Most of them seek to make data collection and communication between various resources easier. Houda et al. In [185] designed a public transport ontology that considers several travel planning concepts. It enables users to select the most efficient path for traveling between locations. Additionally, Bermejo et al. [186] suggested decentralizing decision-making by incorporating an ontology within each vehicle to provide them with emergency reasoning capabilities. To simplify public transportation monitoring, Benvenuti et al. [187], proposed a system based on incorporating ontological key performance indicators (KPIOnto) and transmodel ontologies. (the European reference data model for public transport information systems). Ontology is used to represent and share the meaning of data. Usually, the specific ontology is used to deal with specific problems in a domain that must be constrained. However, although many ontologies are defined, they are difficult to share the meaning between them, especially where the ontologies represent different domains. In reality, the domains overlap. For example, The health monitoring domain overlaps with public transportation, where public transportation affects the health of passengers, and the latter impacts the benefit of public transportation. We believe combining these two fields can be more advantageous for both and will certainly improve people's quality of life. We also believe providing semantic interoperability for data across IoT domains is necessary, and achieving them through these specific ontologies is very difficult. Furthermore, reusing and merging them using alignment techniques to create an all-encompassing and complete ontology is not easy. It is necessary to define ontologies from standard ontologies and follow a standard methodology that enables semantic interoperability across IoT domains.

6.1.4 Semantic web of things frameworks

A set of Semantic Web of Things frameworks has been developed as solutions to achieve semantic interoperability. The first SWoT framework is proposed by Ruta et al. and intends to enhance real-world objects with semantic annotations to facilitate knowledge storage. Theoretically, the method can be used in a variety of situations, but it does not validate or suggest any reasoning based on sensor data [188]. In the following, we present existing IoT-relevant frameworks that support semantic interoperability in IoT systems:

6.1.4.1 FIESTA-IoT(Federated Interoperable Semantic IoT Testbeds and Applications)

The framework is a project that aims to build on the connectivity and interoperability of several IoT platforms and testbeds [122]. It is a Horizon 2020 research project and an innovation action funded by the European Union. FIESTA-IoT provides tools for designing and carrying out experimental workflows, dynamically discovering IoT resources, and accessing data without regard to the testbed. FIESTA-IoT offers IoT Experiment as a service on top of a middleware infrastructure that federates and adapts preexisting IoT testbeds and platforms. These experiments provide various tools and best practices to improve the interoperability of heterogeneous IoT platforms. The FIESTA makes it possible for researchers and experimenters to share and reuse data from diverse IoT testbeds using semantic technologies. The framework has its ontology called "IoT-Fiesta" and makes the alignments with other ontologies.

6.1.4.2 Machine-to-Machine Measurement (M3)framework

The Machine-to-Machine Measurement (M3) framework enables the development of interoperable cross-domain IoT applications [189]. It helps end users and IoT developers semantically annotate M2M data, create cross-domain IoT applications by merging M2M data from various sources, and analyze the data. M3 framework is based on semantic web technologies such as RDF/XML, conforming to the M3 ontology to explicitly explain the meaning of sensor measurements in a uniform manner. The framework represents a common phrase that can describe various abilities that allow connected computers to communicate information and carry out different tasks that require human assistance. In order to transform IoT data from a heterogeneous format into meaningful information, the Machine-To-Machine architecture is employed for semantic annotation and straightforward interpretation. It is used to develop Internet of Things (IoT) applications, help users interpret sensor data, and merge domains.

6.1.4.3 SymbIoTe

SymbIoTe's (symbiosis of smart objects across IoT environments) framework enables a flexible and secure middleware for interoperability across IoT platforms [190]. The primary objective of this framework is to provide an interoperability and mediation framework for collaboration and the federation of vertical IoT platforms. It intends to create cross-domain applications employing different heterogeneous IoT platforms in a coordinated manner. This framework offers an abstraction layer for a "unified view" on different IoT platforms and sensing/actuating resources and seeks to achieve interoperability of IoT systems. It also enables the integration of virtual IoT environments across different IoT platforms and connects numerous APIs. In order to satisfy the trade-off between a desired high level of abstraction and a necessary concretization of particular concepts, symbIoTe provides its own domain ontology for the IoT that is specifically adapted to that constrained solution space.

6.1.4.4 VICINITY

The H2020 VICINITY project offers interoperability as a service " platform for IoT infrastructures that is independent of devices and standards". VICINITY represents a decentralized, bottom-up, standards-based platform that enables the integration of Internet of Things (IoT) ecosystems without having to change their semantics. Its fundamental concept is to create a neighborhood of nodes that can communicate seamlessly with each other on the network [191]. VICINITY improves semantic interoperability by using ontology to represent objects, their characteristics and capabilities, and how to collect information from objects through their own web interfaces, which greatly extends the support for interoperability in the IoT ecosystem.

6.1.4.5 INTER-IoT

INTER-IoT is an open framework that enables seamless interoperability between several platforms at any specific layer or level (device, network, middleware, application, data, semantics). Through an API, the INTER-IoT open framework (INTER-FW) provides a collection of interoperability tools at each layer. It is based on GOIoTP (Generic Ontology for IoT Platforms) ontology [192] and can be used as a solution for any application domain and cross-domains when necessary to make interoperability and enables common interpretation of data and information between them [193]. In order to seamlessly integrate various IoT architectures found in various application sectors, it designed an "interoperable framework architecture". It developed the Inter-Platform Semantic Mediator (IPSM), especially for real-time semantic translations of streaming data. It is considered a Universal syntactic that allows interoperability between different data formats, such as JSON, XML, and others.

6.1.4.6 BIG IoT(Bridging the Interoperability Gap of the Internet of Things)

This project, like INTER-IoT, focused on the interoperability of IoT platforms with an emphasis on revenue via a marketplace. It creates semantic interoperability using the IoT Marketplace and BIG IoT API, a uniform Web API for IoT platforms. The API supports the alignment with the standards developed by the W3C Web of Things group. The BIG IoT API allows service providers on heterogeneous platforms to register resources on the BIG IoT Marketplace. The BIG IoT project is an online platform that links several middleware and platform technologies. It makes use of the concept vocabulary provided by schema.org. The framework is based on a set of tasks to define interoperability [194]:

- Identity administration for resource registration.
- Locate resources using search criteria that are defined by user.
- Access data (download data, publish/record feeds) and metadata.
- Management of vocabulary for concept semantic descriptions.
- Security, including key management, identity management, and authorization.
- Billing enables to generate income via payment and billing processes

Framework	**Semantic interoperability**	**Cross domain**	**Semantic web technologies**						
			Ontology					**SWRL**	**Reasoning Engine**
			Name	**Modularity**	**Reusability**	**Scalability**	**Development Methodology**		
FIESTA -IoT	X	X	Fiesta ontology	-	-	-	SEG3	X	X
BIG IoT	X	X	schema	-	-	-	-	-	-
INTER -IoT	X	X	GOIoTP(Ssn/sosa,)	X	X	X	-	X	X
SymbIoTe	X	X	SSN	X		-	-	X	X
VICINITY	X	X	VICINITY ontology	X	X	-	LOT	-	-
M3 framework	X	X	M3 Ontology	-	-	-	-	X	X

Table 6.3: Comparison of framework semantic web of things

The frameworks that have already been presented shown in Table 6.3 and table 6.4 share some features like their components, application domain, and IoT architectures. The frameworks developed for interoperability problems, especially semantic interoperability, and based and semantic web technologies and defining or reusing ontologies to achieve it. They also benefit from the available IoT technologies like cloud and fog for easing, integrating, and interpreting data semantically and supporting scalability. However, the frameworks have a set of limits that we should consider in our proposed frameworks. They rely mostly on cloud technologies, which made them lose the advantages of other technologies. The developer of frameworks defines their ontologies and avoids reusing a quality ontology, especially standard IoT ontology. The missing methodology for defining these ontologies is due to complicating and incompatible ontologies with one another. Furthermore, the framework focuses primarily on the interoperability of specific fields rather than on a general solution.

Despite these considerable efforts that have been made, the existing framework is based on classical ontologies that cannot handle unclear, incomplete, or imprecise information; for this reason, a fuzzy one must be included [134]. We shall present various research that relies on the use of fuzzy ontologies in the part that follows.

Framework	Application Domains	IoT architecture			
		Devices	Edge	Fog	Cloud
FIESTA -IoT	energy, safety and healthcare	X	-	-	X
BIG IoT	smart parking	-	-	-	X
INTER-IoT	Health, logistic	X	-	X	X
SymbIoTe	smart cities smart residence	X	-	-	X
VICINITY	Energy, eHealth, Living home	X	-	-	X
M3 framework	Touristic Weather	X	-	X	X

Table 6.4: Comparison of framework semantic web of things

6.2 Fuzzy ontology systems to improve the accuracy of semantic Web of things frameworks

The internet of things offers several benefits to our daily lives. It permits minimizing human effort, saving time, effectively using resources, and other benefits. It allows data collection, analysis, and exchange about real-world applications. Most of these data are strongly characterized by heterogeneity and vagueness. The framework semantic web of things proposed to deal with this heterogeneity using the key component ontology. However, this classical ontology has been limited to treating the imprecise and vagueness of data [195]. The ontology component, like the concept, may have attribute values taking vague values. To improve its sufficiency in dealing with imprecise and vague knowledge of classical ontology, Researchers integrated the fuzzy logic principle into a classical ontology for defining the fuzzy ontology. Generally, fuzzy ontology refers to using fuzzy concepts to describe vague knowledge. As classical ontology is usually included in a system that makes a decision. The fuzzy ontology may allow for achieving more accurate results. It is used in the work of Ali et al. [196] for monitoring transportation activities (cars, accidents, movement size, highway situations, etc.). The authors combined it with Semantic Web Rule Language rules (SWRL) to provide passengers with a city-feature polarity map.

As the healthcare domain is characterized by the vagueness and imprecise of data, such works use fuzzy ontology for health monitoring patients suffering from a particular disease. Olaide N. Oyelade [197], for monitoring breast cancer patients, proposes a fuzzy ontology that reduces ambiguity in the classical ontology of breast cancer. The proposal's outcome showed that the ontology had lost a considerable amount of its ambiguity. Shoaib et al. [198] also created and implemented an Alzheimer's disease diagnosis standard fuzzy ontology to alert individuals at high risk of developing Alzheimer's Disease. To enhance the decision-making about early diabetes diagnosis, SAPPAGH et al. [199] proposed the fuzzy ontology-based decision support system. To facilitate using the proposed system, the authors implemented and created an interface that enables the input of patients' data and displays the results of the diagnosis. The proposed system can help physicians when diagnosing patients and provide the patients with decisions.

Achieving semantic interoperability in the IoT domain through handling the heterogeneity challenge is a principal issue, leading to using the semantic web's backbone. The ontology plays a crucial role in semantic interoperability, allowing the representation of knowledge and sharing of meaning. Fuzzy ontology, as an extension of classical ontology, can provide it more efficiency with vagueness and imprecise data. Providing a high-quality fuzzy ontology requires a development methodology that supports the reusability [6].

The previous works define their fuzzy ontology from scratch, where ontologies reuse contributes to the standardization distribution of knowledge, improves ontology quality, and reduces development cost. In the literature, many specific IoT ontologies are defined, and reusing any one of them requires some criteria associated with IoT standard ontologies. The main standardized them are the SSN/SOSA [118], oneM2M [123], and the SAREF [124], which can be used to address semantic interoperability.

6.3 Storing semantic web of things framework output

As the internet of things technology is widely used, many devices are connected to the internet and generate data. To avoid the heterogeneity problem and ensure interoperability, researchers proposed to integrate the semantic web to create the semantic web of things. The latter is based on a set of semantic technologies that allow data representation, process, and exchange.

RDF is the simple and popular semantic technology that provides a graphing model for formally describing Web resources. the semantic web of things framework based on RDF format to represent the output of processing of the huge amount of IoT data. The framework needs an efficient method of storing to structure the data in a manner that enables retrieval of information very quickly. As the RDF format is based on the triple store (subject, predicate, object), many solutions focus on converting the triple store into a column table for benefitting of relational database storage [157] [156] [155]. These solutions also enable information retrieval by converting the SPARQL query to SQL and the inverse. However, converting between the two formats consumes time [200], and the user needs an advanced experience with SPARQL query to make qualitative searches and get accurate results [201]. Moreover, these solutions miss the domain's specificity, where the RDF dataset result consists of distributed files, each file sharing the same syntax with other RDF files [153].

Part II

Proposed approach and results validation

Chapter 7

Proposed Approach

7.1 Introduction

The semantic web of thing framework aims to allow the internet of things domain to exchange data meaningfully to overcome its heterogeneity challenge. It enables billions of devices of the internet of things to collect, exchange data and perform tasks seamlessly. The semantic web of thing framework is based on a semantic web solution for enabling the interchange of knowledge and information. The semantic web is based on ontology, which allows formal and explicit describes the meaning of data, making objects understandable and facilitating their exchange between different systems. However, despite the potential benefits of the semantic web of things, there are still issues related to the exchange of IoT data between different applications and domains, the processing of vague IoT data, and the storage of the results of this framework. This chapter presents our research contributions to solve these problems in the semantic web of things context. Our contributions aim to improve the efficiency and quality of semantic data exchange between domains and applications, deal with vague and imprecise IoT semantic data using fuzzy logic, and improve this framework's storage and retrieval of output using indexing. We also provide application scenarios to demonstrate the real-world applicability of our contributions in particular domains.

7.2 Objective

In the IoT domain, data heterogeneity and interoperability are complex challenges. While the semantic web of things offers a promising approach to solving these problems, but there are still challenges ahead. In this context, we have proposed a framework that achieves the following objectives:

- Used the Edge-Fog-Cloud architecture to enable the SWoT framework to address heterogeneity and interoperability issues by providing real-time processing, scalability, and stock the huge data.

- Enabled sharing and alignment between ontologies to facilitate data exchange between IoT applications and domains. The development of ontology follows the NEON methodology, which extends the standard SAREF ontology. Through this ontology, the framework improves interoperability between different devices and systems and exchanges data that come from several domains.

- Handled the vague and imprecise IoT data by implementing a fuzzification model based on a fuzzy ontology. The framework enabled to provide of the correct processing with accurate results using a fuzzy ontology. The proposed fuzzy ontology is designed following the IKarus methodology and represents a conversion of the classical ontology SAERF. The proposed ontology enabled the framework to exchange and share vague knowledge.

- Addressed the problem of storing and retrieving its output data in the form of BIG RDF data by adding the clustering and indexing layers. The framework used a fuzzy logic to improve the grouping of its outputs into clusters, which reduces the search space through indexing, making it easier to retrieve relevant information. The framework is based on Binary tree with two pivots to ensure the efficiency of quick searching.

- Implement the proposed approach in a real application, such as patient health monitoring for early detection of health problems, reduction of hospitalization, and prevention of adverse outcomes. Application of the proposed approach of health monitoring into public transportation in order to provide passengers with high-quality transportation services and continuous health status monitoring of the patient, and finally, applying the approach in the health monitoring of patients infected with COVID-19 virus.

7.3 The conceptual model of the proposed framework

The proposed framework combines the concepts of the semantic web and the internet of things to enable meaningful interaction between IoT devices and applications. The framework also improved by integrating new concepts such as fuzzy ontology, clustering, and indexing. The framework as depicted in Figure 7.1 can be divided into a set of layers, where each layer has a set of tasks and communication with others. The framework ensures IoT data collection from the environment through the IoT device layer. It offers a semantic layer that enables a shared understanding of data, promotes interoperability, and supports reasoning and intelligent interactions among IoT devices and applications. The Fuzzy layer enables the SWoT framework to improve its capabilities of processing IoT data that is characterized by vagueness and imprecise. This layer improved decision-making across various disciplines, especially the healthcare domain. The clustering and indexing layers are defined for storing the framework outputs and retrieval of this data very quickly. The output of the clustering layer is used in the indexing layer to index similar data to accelerate the insertion and searching time.

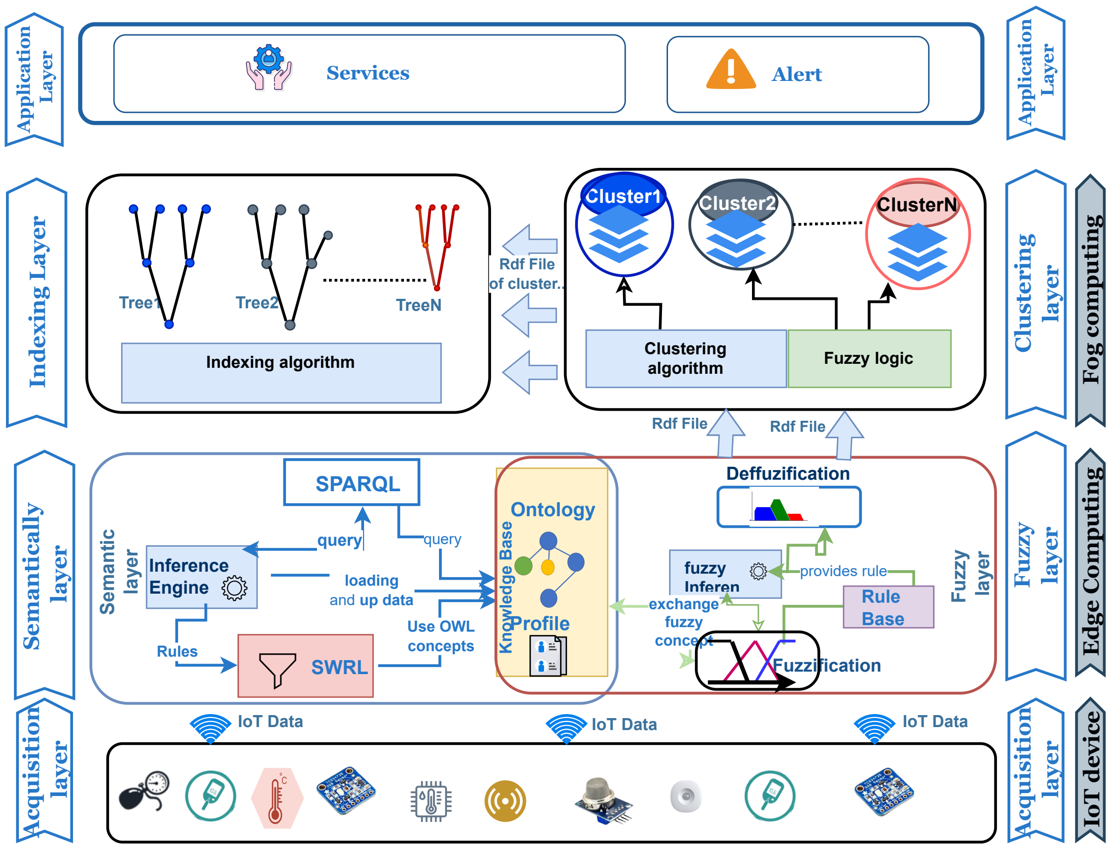

Figure 7.1: The global architecture of the proposed Framework.

The last layer is the services and applications layer. It represents an interface for accessing and requesting the system's services. As these layers are composed of a set of modules, we detail their structure as follows:

7.3.1 Acquisition Data Layer

Many IoT sources are used to collect data from different domains. These data sources represent the set of sensors embedded in objects. These sensors placed on objects collect information about the surrounding environment, such as location, weather conditions, vital signs, etc. The types of data collected depend on their specific purpose and application. The data collected by these sensors are used for various applications including monitoring, decision-making, forecasting, automation, and many others.

7.3.2 Semantic Layer

The semantic layer is responsible for translating raw data into meaningful information. It makes it possible to represent, interpret and deduce new knowledge. It ensures data consistency, supports decision-making, and facilitates interaction with users. The semantic layer mainly consists of a set of modules that perform various tasks, These modules include:

7.3.2.1 Knowledge base

A knowledge base is intended to allow the description, organization, and sharing of information in order to facilitate its analysis and reuse. It allows the representation of the raw data acquired by the sensors by defining the concepts and the relations between them (ontology). In addition, it may include other information in the form of RDF files associated with the application domain, user profiles, etc. This information helps to enrich the ontology and to infer new knowledge.

7.3.2.1.1 Ontological module This module presents an ontology that provides a structured representation of information based on specific elements such as objects, relationships, properties, and other elements. It enables to model of the relationships between the concepts of the domain covered by the ontology, indicating how they are related. These concepts are chosen from the knowledge domain in a way that considers the specific objectives and scope of the ontology and the needs of the end-users. This will ensure that the ontology is well-defined and accurately represents the domain concerned.

7.3.2.1.2 Profile Module The profile module is a file that stores information to manage the profiles of users or other entities in the system. The profile module is in RDF format and has an appropriate structure with the ontology to facilitate access to this data. Typically, it contains information that can enrich the ontology and facilitate understanding of the system, such as contexts, attributes, categories, or other relevant elements. For example, The patient profile module can contain medical profiles, personal information, diagnostic history, etc.

7.3.2.2 SWRL module

The knowledge base contains an ontology that plays a crucial role in data representation but cannot express all relationships, especially hidden ones. The SWRL module is based on the Semantic Web Rule Language (SWRL), expressed using the terms of the ontology concepts (classes, properties, individuals). It allows the framework to validate relationships between ontology concepts and to perform inferences on the ontology to create new information. As the module is based on SWRL rules, It is essential that defined by experts in the domain-specific to guarantee that they accurately reflect domain-specific requirements. Additionally, a reasoning engine must be used to accomplish ontological inference.

7.3.2.3 The inference engine module

The inference engine module allows utilizing the knowledge base, including the ontology, module profile, and SWRL rules, to create new relationships to grow the ontology and ensure consistency. This tool can infer logical conclusions from a set of facts, supporting deducing significant information from the semantic sensor data. To semantically evaluate the IoT data, We can employ one of the available reasoners, such as Pellet, Hermit, etc. The module is also enabled to enrich the results of the user's query with new information.

7.3.2.4 Query engine module

The query engine module uses the standard query language SPARQL (Simple Protocol and RDF Query Language) to create queries, explore the semantic descriptions and discover IoT resources. The module handles queries received from the application layer that present a request of users to get a service and interesting information.

7.3.3 Fuzzy layer

The fuzzy layer is a layer that enables integration with the semantic layer to improve the accuracy of its results by handling vagueness and imprecise data. The layer consists of modules that combine with the semantic layer module. Every module in the layer produces an output that another module uses as input, ensuring that the framework processing operates correctly. Most modules are the following:

7.3.3.1 Fuzzy ontological module

The Fuzzy ontological module represents an ontology that enables to express and share knowledge that might be vague and imprecise. The module is a transformation of the ontological module of the semantic layer to improve its representation and sharing. It aims at improving reasoning about imprecise and uncertain knowledge, which can be useful in many domains where boundaries are vague, and information is imprecise.

7.3.3.2 The Fuzzification module

The fuzzy module converts the collected IoT data to fuzzy variables by providing graduated truth values using membership functions. The latter represents a key element, it is used to give a degree of membership to each concept or relationship in the ontology. The membership function can be determined based on various methods, such as human expertise, system standards, etc.

7.3.3.3 The fuzzy inference module

The fuzzy inference module is the third module of the fuzzy layer, based on fuzzy if-then rules. The Mamdani engine is a type of this inference that builds the relationship between input data and output. The structure of this engine is simple and supports multiple input and single output systems, as well as multiple input and multiple output systems. It is best suited for the recommender system.

7.3.3.4 The Defuzzification module

Defuzzification module is the reverse process of fuzzification module. It provides a concrete, usable result from the truth values. The module is run after the input data has been fuzzified and fuzzy logic operations, such as inference, are finished. The outcomes of this module support making specific decisions or taking specific actions.

7.3.4 Clustering layer

The clustering layer organizes and groups the output into a cluster. It separates the data into more manageable, smaller groupings depending on certain criteria, such as a range of values or individual property. The clustering layer facilitates the searching and indexing processing. In addition to the algorithm or system used to partition the data, the cluster layer also enables to use a fuzzy logic to improve classification by handling uncertainty and ambiguity in IoT data.

7.3.5 Indexing layer

The indexing layer consists of techniques for organizing and managing large databases to improve access times and reduce search space. The layer benefits from the clustering layer by using the characteristics and results of the clustering for indexing. As many different indexing techniques are available, selecting ones is associated with criteria like the size of data, type of data, and others. The binary tree is a simple tree-based structure in which nodes can have no more than two children. The tree can also be improved to provide more capabilities in indexing. For example, adding two pivots to each node ensures that the tree remains balanced, improving performance during the search, insert and delete operations.

7.3.6 Services and applications layer

The services and applications layer is responsible for addressing the needs of users in various application areas. It provides users with an interface to help them access and request the system's services. This layer presents the final result of the processing of the framework and enables communication with users using different techniques, such as messages, email, etc.

7.4 IoT architecture of the proposed framework

The semantic web framework aims to tackle the data heterogeneity problem, achieve semantic interoperability, produce new knowledge by using reasoning, enable making decisions, and improve the services and applications that are based on the Internet of Things. The semantic web framework to achieve these goals needs adequate storage and processing capabilities to effectively manage semantic IoT data. Nevertheless, Most semantic web of things frameworks are cloud-based. This model has difficulty related to latency, cost, etc. To overcome these challenges, the framework needs an architecture that allows IoT data processing to be provided depending on processing requirements and ensures latency and scalability. The Edge-Fog-Cloud [202] is a distributed computing architecture model that distributes data processing, storage, and management functions across multiple levels to improve the latency, scalability, and quality of service of Internet of Things (IoT)-based applications. As the framework consists a set of layers, it enables to decentralize with Edge-Fog-Cloud architecture. The devices layer, the edge layer, the fog layer, and the cloud layer are the four layers of our model shown in Figure 7.2. The model categorizes the computing devices based on their capabilities and the requirements of the proposed framework. The proposed model is as a following:

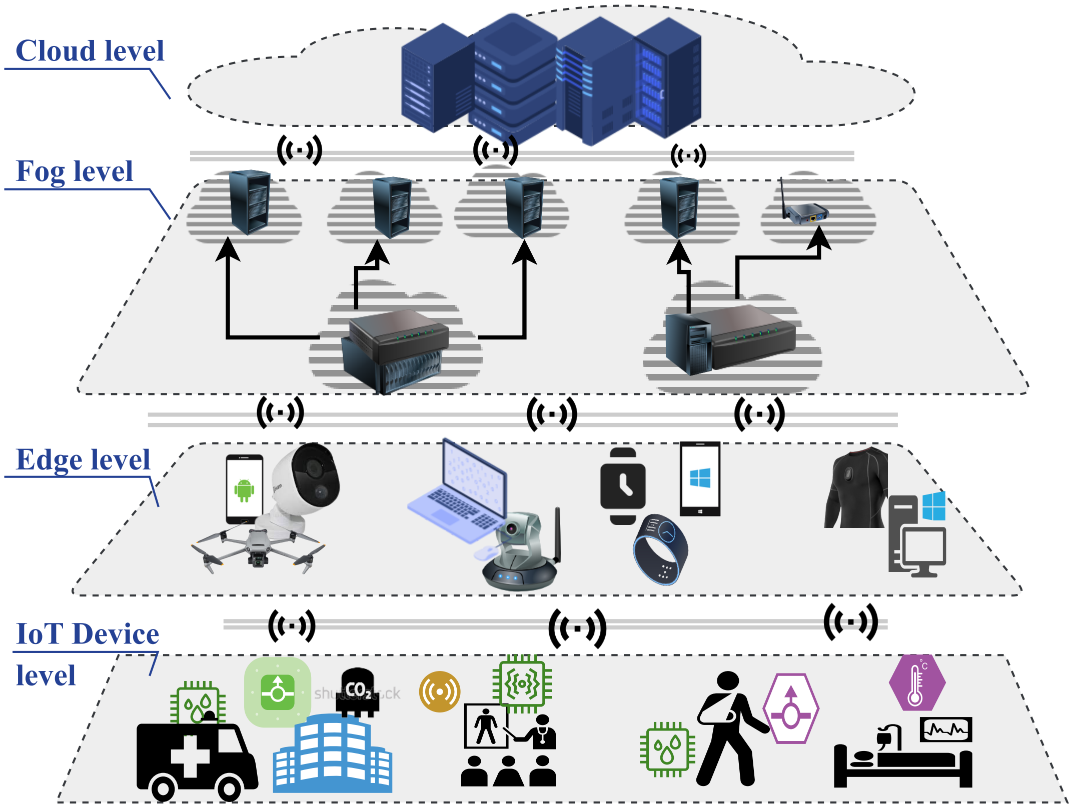

Figure 7.2: Edge fog cloud architecture in the healthcare domain.

7.4.1 IoT devices level

It is the lowest level. It consists of all IoT devices which incorporate objects such as sensors, actuators, or sensing devices that can be measured and collect data. Sensors are devices embedded in connected objects to collect data about their environment. The sensors are of different types: physical sensors (temperature sensors, light sensors, etc.), biometric sensors (blood pressure sensors, oxygen sensors, saturation sensors, etc.), position sensors (GPS), and environmental sensors (water sensors, air quality sensors, etc.). The choice of sensors in this layer is depending on the implementation of this framework. For example, in healthcare, biometric sensors that can measure the patient's vital signs and can be integrated into wearable devices. The data collected from the sensors can also be exchanged over the internet for processing and interpretation. IoT devices use a number of communication technologies to exchange data, such as Wi-Fi, Bluetooth, etc. It also consists of actuators that support action. Actuators enable actions to be carried out, like activating an alarm, opening a door, etc. In healthcare, for monitoring patients, a set of sensors working in synchronization to measure patients' vital signs can be equipped in wearable devices. The wearable sensing devices are characterized by their low cost and ease of use and come in the form of a smartwatch, glasses, helmet, gloves, etc. The wearable can be working as an edge for processing the gathered data.

7.4.2 Edge level

It is the second level of the architecture, which includes the semantic and fuzzy layer of the proposed framework. It enables to process the collected data close to the source data without the need for sending all data to cloud. The level represents wearable or other smart items that can process and store data, such as smartphones, smart wearables, and others. This level has capacity storage and processing allows to provide a real-time processing of IoT data. The level uses the ontology, SWRL rule, engine reasoning for semantic processing or fuzzy ontology, fuzzification, and fuzzy inference to provide a result more accurate. It increases the efficiency of data processing and decision-making. Due to its restricted processing and storage capabilities, the edge level can't always give the necessary processing and deliver the right outcomes.

7.4.3 Fog level

The fog level is able to carry out tasks that require more capabilities and information since it has a larger processing and storage capacity than the edge level. Additionally, it offers edge level output temporary storage. The fog level also covers a broader region than the edge, enabling communication and information sharing with several edge devices. It incorporates the clustering and indexing layers of the proposed framework, for simplifying data storage and retrieval. The fog level improves the efficiency of data management and makes it easier to retrieve and use the necessary information, thus contributing to better exploitation of cloud resources.

7.4.4 Cloud level

The top level in the architecture is typically the cloud layer, which is responsible for storing and processing large amounts of data generated by the lower levels. After classifying and indexing the data in the fog layer, the cloud layer store data for analysis and processing, especially in complicated cases and complex treatments. The architecture lowers the cost of setting up the infrastructure by giving tasks enough resources. Additionally, this architecture can provide data enrichment to guarantee accurate processing, even in complex scenarios. Using edge computing, fog computing, and cloud computing enables improved the proposed framework performance, while a quality ontology definition enables good results.

7.5 Knowledge representation: ontology

An ontology is a common, standardized language that allows knowledge to be represented by defining concepts and relationships in a specific domain. An ontology is used to facilitate understanding and interoperability between different applications and systems. For defining an ontology, there exist various ways, but developing a quality ontology is a complicated process that requires rules and consensus. The quality of the ontology and the ability to share data with others are greatly impacted by this process.

On the other hand, methodologies play a crucial role in ontology building, as they provide a set of guidelines for the development process, life cycle models, techniques, and tools. Neon methodology is a scenario-based approach that enables reusing, possibly re-engineering knowledge-aware resources, using alignments, exploiting ontological and non-ontological sources, and other techniques. The ontology development follows the Neon methodology to ensure standardization of ontology definition, which enables clear and consistent ontology definition, reusability, sharing and alignment, in addition, to accepting maintenance and scalability. The Neon methodology supports the reuse of other ontologies to develop the ontology, which reduces development costs, improves the quality of existing ontologies, and allows the developer to benefit from the experience of others. For these reasons, the methodology offers scenario 3 (see section 3.4.7.2) to enable ontology resource reusing. The scenario proposes different ways to reuse ontological resources, taking into account the different requirements that can be defined with a set of questions related to the IoT domain. Examples of these questions:

- What is the internet of things resources?
- What are the components of the internet of things?
- What is the purpose of the internet of things
- What are the services provided by the internet of things?
- How to communicate internet of things resources?
- What are the restrictions of the IoT domain?
- And others.

For reusing the ontology, we follow a process defined by this scenario: (1) we look for ontological resources by search engines using the terms internet of things ontology, standard IoT ontology, and the previous questions. (2) (3) by evaluating the ontological resources according to the answers to the previous questions and the result of the comparisons of tables 6.1 and 6.2 against a set of criteria (reusability, modularity, support alignment), (4) we have chosen the SAREF ontology as the most useful ontology resource for our project. The ontology is characterized by its moderate complexity and modular structure, which means that a modular element can be reused or updated without modifying the whole ontology.It also supports alignment with other ontologies, such as oneM2M, to facilitate the exchange of information.

As SAREF ontology identifies concepts represent IoT domain and connecting them with relationships, we reuse some classes of this ontology and their subsets. We reuse Devices class with their subset Sensors, Actuator. The classes *Measurement, Feature-of-interest, Property* and *unit-of-measure* are also selected. In addition to the object properties that define the relationship between these classes such as *measured-By, consist-Of, make-measurement, relate-To-Property, is-Measured-In*, and data types such as *has-Value, Has-Timestamp*. The ontology has been expanded to include a collection of wearable technology, environments, smart cities, and energy ontologies. The SAREF ontology is expanded upon in SAREF4wear for representing wearable devices by using the same concepts and properties with adding a new little concepts such as *wearer, wearable devices, etc.*

Classical ontology models might be unable to manage some of the knowledge related to the IoT domain, since it is vague or imprecise. Researchers have determined that creating a fuzzy ontology is the most effective solution to this challenge [128]. Converting a classical ontology to a fuzzy ontology presents a shortcut to optimizing time, cost, and effort [203]. The IKARUS-Onto methodology is a unique approach that expands classical ontologies to fuzzy ontologies. This methodology ensures that the creation process of fuzzy ontologies is correct, shareable, and reusable. To avoid the problems of sharing and alignment of the new fuzzy ontology and benefit from reusing a SAREF ontology, we use he IKARUS-onto to create a fuzzy ontology from the classical ontology. The IKARUS-onto is methodology consists of a number of steps, beginning from the acquisition of the SAREF ontology and ending with the delivery of a high-quality fuzzy ontology. The IKARUS-onto consist presenting the need for fuzziness step that enables to determine some cases of vagueness that need to be determined and tackled.

As the SAREF ontology has standard concepts that miss fuzzy cases, the conversion process will start when the ontology is enriched by new concepts related to the ontology's domain. Furthermore, to evaluate a proposed framework's effectiveness, it is crucial to implement it in real applications to produce concrete and measurable results.

7.6 Application of the proposed semantic web framework in public transportation and healthcare

In recent years, the new paradigm internet of things by connecting devices to the internet and sharing their collected data-enables the change of traditional life into a high-tech way of living. It is integrated into different domains with the objective of making them smarter. It allows the development of the traditional tools, management, and mechanism that contribute to creating smart domains like smart industries, smart agriculture, and others with more capabilities associated with collecting, exchanging, and processing data in a way that allows working autonomy.

The Internet of things supports healthcare services by collecting data about a person's health and sending it in real-time to hospitals or physicians to provide the best decisions at the right time. It improves the daily work in hospitals and ensures continuous monitoring of patients, especially those who are aging or have chronic diseases, through the availability and sharing of their data. The Internet of things increases the interaction between physician and patients, enhance health services, and provide more protection to patients in all facets of life. As a patient's health is influenced by the quality of food and the type of activities of patients, it is also influenced by public transportation. Public transportation allows passengers to move from one place to another to perform their activities. IoT allows agencies to collect data from the means of transport and easily exchange and process it in real-time to provide new information to passengers or these transportation means. As millions of people use Public transport daily, even for long hours, it can play a crucial role in improving the performance of health systems. Equipping public transportation vehicles with advanced sensors and passengers with wearable devices that collect real-time information about passengers' health allows for an early diagnosis. Integrating health monitoring with public transportation allows for the utilization of passenger health data to enhance the overall effectiveness of the healthcare system. It enables an efficient diagnosis, reduces efforts and costs and avoids pressure on hospitals by providing health care centers with real-time information on the health status of passengers [13]. Integration of these two domains requires an IoT architecture to enable ease it.

7.6.1 Modelling of Integrating Health Monitoring in Public Transport using the proposed IoT architecture

Internet of things technologies play a crucial role in combing public transport with health monitoring. It provides them with various objects that can generate and exchange data and then process it to provide services to end-users through applications. Most applications in the public transportation or health monitoring sectors are cloud-based, but this model has difficulty providing real-time patient monitoring, which is a critical need in the monitoring of passengers. Modelling of the integration of both domains using the proposed framework enables the management of distributed data, including that produced by the Internet of Things devices. The model of the proposed framework enables low latency, storage capacity, scalability, and adopted the requirement of this integration that helps reduce costs and provide improved and reliable transport health services. The four layers of the proposed framework in the integration of healthcare with public transportation that depicted in Figure 7.3 will be as follows:

The IoT device level refers to a set of sensors mounted on vehicles or at bus stops that gather information about the surrounding environment, such as the location, the number of passengers, the condition of the vehicle, etc. It also consists of actuators that support action. Actuators enable actions to be carried out, like activating an alarm or opening a car door. To allow health monitoring of passengers, wearable devices consist of a set of sensors working in synchronization to measure passengers' vital signs. Wearable sensing devices are characterized by their low cost and ease of use and come in the form of a smartwatch, glasses, helmets, gloves, etc.

The Smart Bus Stop (SBS) is the most crucial component. The smart bus stop works as an Edge of computing. It provides real-time data processing generated by connected devices associated with public transport and health monitoring through processing it close to its sources. The smart bus stop analyses and processes the data and sends an immediate response to passengers, vehicles, or to anyone who needs this information. It helps to make quick decisions because the smart bus stop is very close to the connected devices. Through the Media Access Control address of a passenger's wearables, the SBS can identify and connect with the passengers that wear them. It also utilizes wireless communication protocols to obtain data from mobile devices and vehicles that have sensors. It aggregates additional information by gathering data from other bus stops or high-level transport managers for more precise processing. Despite the smart bus stops communicating and exchanging data with their neighbors, they have a limit in covering public transport in large areas such as small cities, which necessitates additional processing data and storage capacity.

In transportation geography, many small bus stations are distributed in a small city to enable seamless public transportation management. Small bus stations provide passenger services and contribute to managing vehicles and other transport means. Using the same principle in the third level, the small bus station is considered a fog node to enable computation, networking, and data management of data collected by many connected devices. A fog node enables low latency and reduces bandwidth by avoiding transferring every bit of information to the cloud. It obtains and aggregates the data from multiple sources like a smart bus stop, devices, and other sources. It temporarily stores passenger health data released by the smart bus stop to avoid repeating the process, and can pull data from the cloud. It bridges the gap between the transport agency and smart bus stops.

The transportation agency considers as a cloud. The transportation agency provides to do complicated calculations and long-term storage of stored, processed data on cars, road, smart bus stops, and passenger medical profiles. Transport agencies can use big data tools to manage and analyze the collected data and discover new knowledge to support decision-making. It can exchange passenger data with a hospital to check or detect a new disease, such as a pandemic, by observing a global change in patient data in a region or area. The transportation agency uses the various data gathered to raise service quality and simplify management. For instance, the system can estimate and shorten the waiting time based on passengers' arrival and departure times at a bus stop.

The modelling architecture able to incorporate additional sources like wearable devices, smart bus stops, and bus stations. It focuses on providing the processing close to the devices that generated the data, reducing the network load and achieving low latency. Additionally, when one server malfunctions, it has no impact on the other servers and keeps the system from shutting down entirely. By providing tasks with adequate resources, the architecture reduces the cost of setting up the infrastructure. This modelling is more beneficial for the public transportation sector because it enables them to fully utilize the various technologies that are provided by this architecture.

Although the modelling by closing the processing near IoT source data helps reduce the high level of heterogeneity, managing this heterogeneity that characterizes the data generated by large IoT devices connected requires another architecture systematically.

7.6.2 Modelling of Integrating Health Monitoring in Public Transport using the proposed framework

This section explains and describes the concept of the system, which enables the different components of the proposed IoT architecture like smart bus stops, bus stations, and

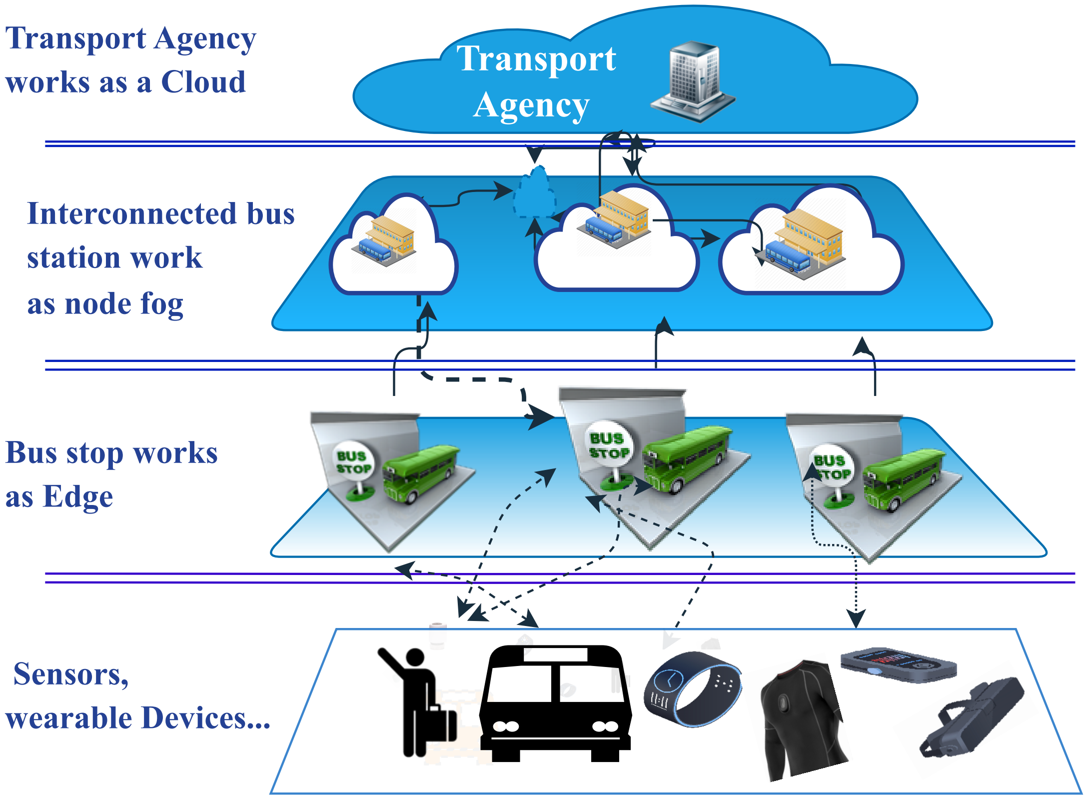

Figure 7.3: Modelling of Integrating Health Monitoring in Public Transport using the proposed framework architecture [13].

transportation agencies to monitor passenger health and provide valuable recommendations or transportation health services. As the proposed framework consists of a set of layers where each layer also has modules, in the following, we detail the essentials of them the semantic layer and service and application layer.

7.6.2.1 Ontological module

Integrating public transportation and health monitoring covers many concepts related to diagnosis, treatment, and patients (disease, drugs, state, symptoms, etc.). Also, it consists of concepts about public transportation such as vehicles (car, ambulance, motorcycle, etc.), travel, bus stop, and others, and concepts related to the internet of things like sensors, actuators, etc. Representing and explaining the relationships between all these concepts require a formal model called ontology. It is based on specific information such as objects, relationships, properties, and others to enable information representation.

SAREF ontology is the ontology of the proposed framework. The ontology has a set of extensions, such as SAREF4Wear [204] is defined to represent the wearable devices in the IoT domain. To define an ontology enables to represent the health monitoring of passengers in the transportation public, the SAREF4Wear ontology needs improvement and enrichment. Based on this methodology, we pass through two steps to develop the ontology representing the integration of health monitoring systems with public transport data. In the first step, we follow scenario one, "Building ontology networks from scratch without reusing existing resources" by developing a new ontology for public transportation. We focus on our architecture based on the smart bus stop. In The second step, we followed scenario five, where we merged the two previous ontologies (as shown in the Figure 7.4) to design a new ontology called passenger transport monitoring (PTM ontology).

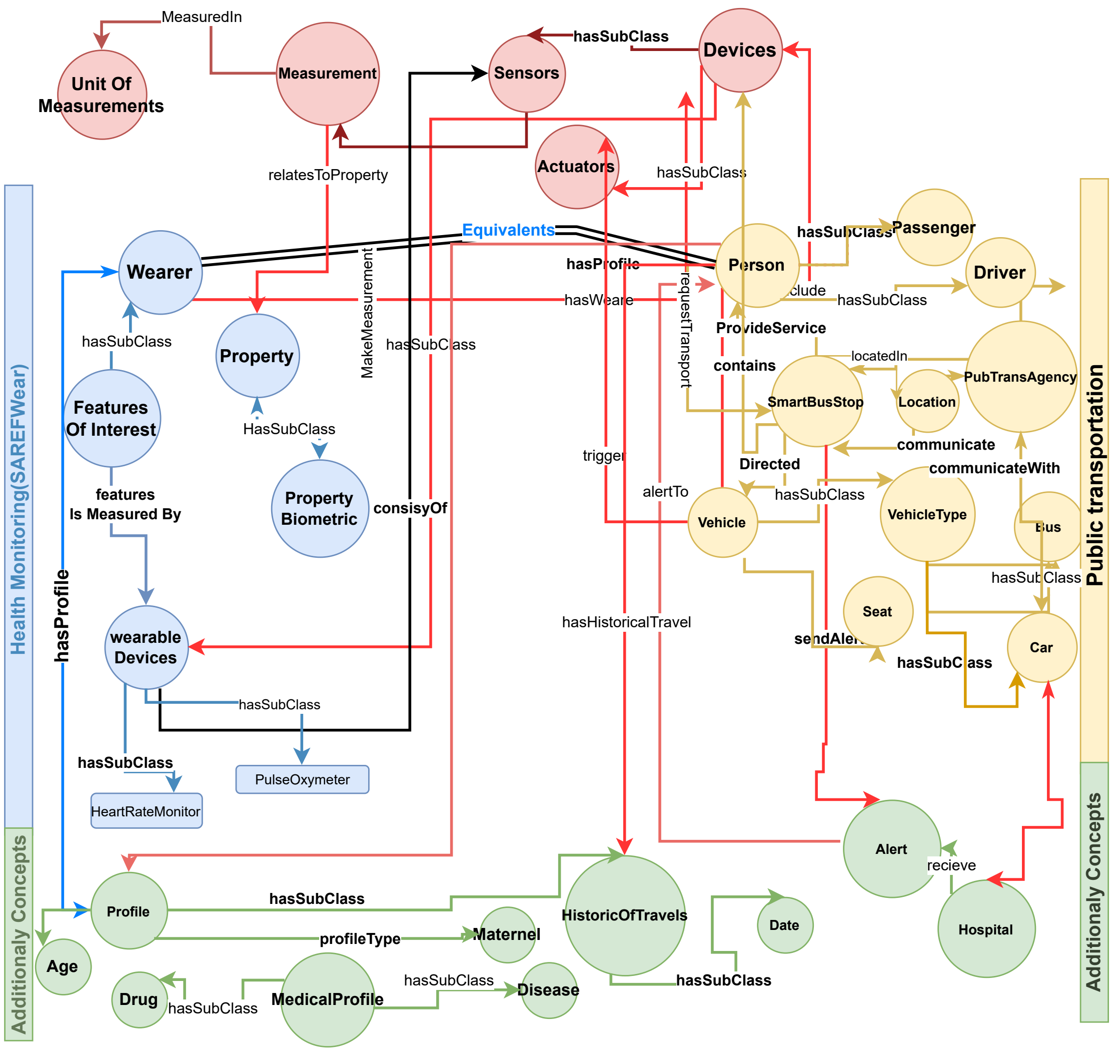

Figure 7.4: Ontological scheme for monitoring health passenger in public transportation.

7.6.2.1.1 Smart Public Transportation Ontology

IoT data can be gathered and communicated from various public transportation sources. Nevertheless, their heterogeneity and lack of formality make sharing and reusing their meaning difficult. The ontology enables us to annotate and describe public transportation resources connected to IoT systems, such as smart bus stops, buses, and other vehicles. The ontology satisfies the domain's requirement for a common language by responding to fundamental queries and other inquiries about public transportation:

- What is the domain of ontology?
- What is the purpose of ontology?
- What is the use of ontology?
- What are the resources of public transportation?
- Who are the users of public transportation?
- What are the services provided by public transportation?
- How do we communicate the resources of public transportation?
- What is the role of IoT in public transportation?
- How many passengers in the bus?
- Where are the destinations of passengers?

Through the response to these questions and others, we can define and represent the concepts as a class or subclass that clarify in the following Table 7.1:

In order to ease the merging between this ontology and the SAREF ontology, we defined the terms that express public transport domain concepts similar to those used in the SAREF ontology, such as *Device, sensors, actuators Measurement, Unit-Of-Measure, and property*(see Table 7.1). Two kinds of properties define ontology; object properties present the relationship between the classes, and dataType properties present a connection between the instances of the class (individuals) and the value. In table 7.2, we present some object properties and their domain and range:

Table 7.3 also illustrates the data properties with their domain and range witch defined to capture information about a passenger and vehicle.

Figure 7.5 shows classes, attributes, and individuals are used to build the proposed ontology that enables sharing a common understanding of heterogeneous data in the public transport domain using IoT. The ontology also facilitates knowledge reuse and analysis,

Concept	**Explication**
Person	represents the public transport user. The class has other subclasses such as Passenger, Driver
TransportAgency	Represents the agency responsible for managing public transport resources and services.It includes the subclass of small bus stations that manage public transport in a small area or in a few streets
SmartBusStop	Represents a location along a route where vehicles stop to pick up or drop off passengers. This concept is linked to many properties
BusStation	Represents a small agency for managing public transport
Vehicle	Describes the means of transport. It has several sub-classes, for example, the Type-of-Vehicle, to represent the mode of transport.
TransitCalendar	The service's operational hours and dates are specified in the transit calendar.
Transit Line	A transit line is a permanent alignment that houses and serves vehicles that run according to a set timetable.
Reaction	Includes action taken by the driver through the smart bus stop recommendations like acceleration and changing the line.
Trip	Irefers to the journey taken by a passenger using a mode of transportation that is available for use by the general public.
Seat	Use by passenger to take place
Device	Physical objects that perform a task in public transport. It includes subclasses of sensors and actuators
Sensors	Represent the sensors used in public transportation means like infrared sensors
Actuators	Represent the actuators equipped in public transportation means for making some actions.
Measurement	Represents an operation performed by devices on a property to obtain its value (measurement value).
Unit-Of-Measure	the unit of measure of sensor value.

Table 7.1: Set of the concepts as a class for representing public transportation

N	Object Properties	Domain	Range
1	RequestTransport	Passenger	SmartBusStop
2	ContactTo	SmartBusStop	Vehicle
3	Manage	TransportAgency, BusStation	SmartBusStop
4	ProvideServices	SmartBusStop	Passenger
5	getInformationFrom	Passenger	Calendar
6	Compose	Trip	SmartBusStop
7	LocatedIn	Passenger, vehicle, SmartBusStop	Location
8	BeniftService	Passenger	Service
9	EquippedIn	Sensors	Vehicle
10	MackeMeasurementBy	Sensors	Passenger

Table 7.2: Object properties ontology public transportation

N	Data Properties	Domain	Range
1	SmartBusStopName	SmartBusStop	String
2	hasAge	Passenger	Integer
3	NumberofPassengers	Passengers	Integer
4	lineNumber	Route	Integer
5	SwitchOn/Of	Airconditioned	Boolean
6	hasCapacity	Vehicle	Integer
7	hasValue	Measurement	String
8	hasTimeDate	Measurement	Time Date
9	SeatNumber	Seat	Integer
10	StreetName	Street	String
11	hasAdressMacValue	Wearable devices	String

Table 7.3: Data properties ontology public transportation

allowing seamless vehicle management and providing passengers with appropriate information and services in the shortest time and at the lowest cost.

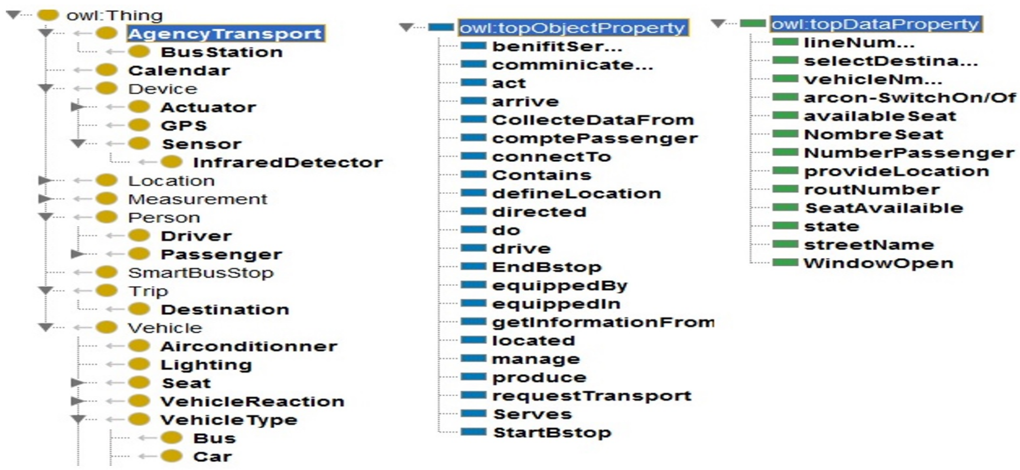

Figure 7.5: Proposed smart public transportation ontology

7.6.2.1.2 Passenger Transport Monitoring Ontology

To develop the passenger transportation monitoring (PTM) ontology, we follow scenario number five. We reuse and merge the SAREF4wear ontology that represents health monitoring using wearable devices with our ontology that was developed to represent smart public transportation. In addition, we added new classes to enrich the ontology by providing more presentations on the integrations of two domains. The construction of the public transportation ontology, whose structure is somewhat similar to that of the SAREF ontology and shares several concepts within, makes it easier to unite the two ontologies.

Some of the classes from the two prior ontologies and their relationships and data types have identical terms. To address the issue, the classes have identical terms, such *devices, measurements, and unit of measurement, etc*, were combined into a single class along with all their subclasses, data relations, and attributes. We also established new connections between classes from other earlier ontologies, such as the *equivalentTo* connection between the *Person* class from the Public Transport ontology and the *Wearer* class from the SAREF4wear ontology. To better represent the knowledge of both domains, we enrich the ontology with new concepts. Table 7.4 presents a sample of a new addition to merging of previous ontologies.

New identifier	Categories	Explication
Hospital	class	Hospital give/get some information about passenger
MedicalProfil	class	contains information about passengers with their subsets of classes, such as Disease, Medication, etc.
Alert	class	Message or notice sent to passenger or hospital
HistoryOfTravels	class	Contains information about the history of trips of passenger
GetInformation	Object Properties	The smart bus stop gets information from the medical profile.
hasMedicalProfil	Object Properties	Relate passengers to medical profiles.
IsSmoking	Data Type	Checked if the passenger had smoked or not.
hasdisease	Data Type	Checked if the passenger had a disease to help in the diagnosis.
AlertInformation	Data Type	Message of Alert.

Table 7.4: Enrichment of MTP ontology by adding classes and properties

7.6.2.2 Profile Module

The system needs more information than the vital signs collected from the sensors to diagnose passengers' health status and make accurate decisions. The profile module is a semantic file containing information about medical profiles, personal information, history of diagnosis, travel history, etc. The medical profile is an essential module that includes information about the passenger's health status, such as previous illnesses, symptoms, smoking habits, family history, and treatments. The medical profile refers to information on the history of diagnoses and some exceptional services provided to passengers during his/her past travels. The personal profile refers to common passenger information, such as Gender, Age, Job, location, height, weight, and others. The semantic file can be stored in a wearable device such as a smartwatch, smartphone, or other devices with a short memory and can communicate to the internet. The physician or hospital can access the file for consultation or add other instructions.

7.6.2.3 SWRL module

In the following, we present some rules written by SWRL that represent most cases that will treat by the system and can be expressed in terms of OWL concepts to improve the health passenger transport monitoring process.

R0: *Person(?p), SmartBusStop(?SBS), requestTansport(?p, ?SBS) -> Passenger(?p)*
This rule is used to present the relationship between a person and a passenger. If person P asks for the smart bus stop as a means of transport, the system considers him as a passenger. Similarly, we define the person who drives the vehicle as the driver.
R1: *Person(?p) , Vehicle(?v) , drive(?p, ?v) -> Driver(?p)*
R2: *Passenger(?P) , SmartBusStop(?SBS) , requestTransport(?P, ?SBS) , Vehicle (?V) , connecte(?SBS, ?V) , InfraredDetector(?I) , containsDevice(?V, ?I) , infraredDetectorValue(?I, ?vl) , swrlb:lessThan(?vl, 30) -> hasCapacity(?V, true) , provideServices (?SBS, ?P) , benifitService(?P, ?V)*

Rule 2 represents the status when the smart bus stop (SBS) receives transportation requests and the next vehicle does not reach its capacity; the system allows the passenger's number to increase and provides the passenger with transport service.

R3: *SmartBusStop(?SBS) ,Vehicle(?V), recommand(?SBS,?V), Reaction (?R) ,ReactionName(?R, "change your line to x line")-> makeReaction(?V,?R).*
The bus stop, bus station, and agency transport manage the vehicle and recommend a vehicle's reaction like changing the line.

R4: *Passenger(?P) , WearableDevices(?wd) , featuresIsMeasuredBy(?P, ?wd) , Sensor(?S) , consistOf(?wd, ?S) , Measurement(?M) , makeMeasurement(?S, ?M) , Property(?Pr) , relatesToProperty(?M, ?Pr) , hasValue(?M, ?V) , swrlb:greaterThan(?V, 96) , swrlb:lessThan(?V, 98) , SmartBusStop(?SBS) , requestTransport(?P, ?SBS) , collectDataFrom(?SBS, ?wd) -> provideServices(?SBS, ?P).*
The smart bus stop before offering any service to passengers. It obtains and analyzes wearable device measurements to evaluate their health status. Example: P= Passenger6, wd= PulseOxymeter, S = photoDetctor, Pr =OxygenSaturation, V=97 hasValue, SBS=BusSTop3 -> provideServices (BusSTop3, Passenger6).

To allow decision-making related to the patient's health state through SWRL rule, the following rules will write according to the MEWS system.
R5: *Passenger(?P) , SmartBusStop(?SBS) , requestTransport(?P, ?SBS) , WearableDevices(?W) , collectDataFrom(?SBS, ?W) , featuresIsMeasuredBy(?P, ?W) , Sensor(?S) , consistOf(?W, ?S) , Measurement(?M) , makeMeasurement(?S, ?M) , Property(?Pr) , propertyName(?Pr, "SPO2"), relatesToProperty(?M, ?Pr) , hasValue (?M, ?vl) , swrlb: greaterThan(?vl, 96) , Services(?S)-> provideServices(?SBS, ?P) , ServicesVehicle(?S, 'Bus').*

The passenger uses their wearable device to measure their vital signs. It may get the measurement value of Systolic BP value between 100 and 199, or/and body Temperature value between 36 and 37.9 or/and Respiratory rate value between 9 and 20, or/and heart rate value between 60 and 100, or/and oxygen saturation value between 95 and 100. The diagnosis results of the system showed the health status of the passenger is normal, and then the smart bus stop SBS offers regular service. The smart bus stop (SBS) provides this passenger with regular services.

R6: *Passenger(?P) , SmartBusStop(?SBS) , requestTransport(?P, ?SBS) , WearableDevices(?W) , collectDataFrom(?SBS, ?W) , featuresIsMeasuredBy(?P, ?W) , Sensor(?S) , consistOf(?W, ?S) , Measurement(?M) , makeMeasurement(?S, ?M) , Property(?Pr) ,propertyName(?Pr,"BodyTemperature"), hasValue(?M,?v) , swrlb:greaterThan(?v,39) , Vehicle(?V) ,SeatKind(?KS) ,include(?V,?KS) ,Airconditionner(?Ac), include(?V,?KS) , connecte(?SBS,?V), Services(?S) -> provideServices(?SBS,?P) , ServicesVehicle(?S, "Bus"), ArcSwitch(?Ac, true).*

According to the table of MEWS, if the measurements of passenger P include one or two vital signs value belongs to the column of the score" 2", or three of them belongs to the column of the score" 1", the passenger his/her health is abnormal, then the smart bus stop SBS offers a specific service. For example, a seat with a switch to start the air-conditioner and a message to notify it.

R7: *Passenger(?P) , SmartBusStop(?SBS) , requestTransport(?P, ?SBS) , WearableDevices(?W) , collectDataFrom(?SBS, ?W) , featuresIsMeasuredBy(?P, ?W) , Sensor(?S) , consistOf(?W, ?S) , Measurement(?M) , makeMeasurement(?S, ?M) , propertyName(?Pr,"BloodPressure") , relatesToProperty(?M, ?Pr) , hasValue(?M, ?v) , swrlb: greaterThan(?v, 180) , Vehicle(?V) , connecte (?SBS , ?V) , Hospital(?H) , communicate(?SBS, ?H), Alert(?A) , Services(?S) -> provideServices(?SBS, ?P) , ServicesVehicle (?S,"emergency car") , sendAlert(?A, ?H).*

If the measurements of passenger P show that any of its vital signs value belongs to the column of score" 3" in the table of MEWS or three of them belong to the column of the score" 2", the passenger his/her health is in danger; the smart bus stop (SBS) provides an emergency car and immediately alerts the hospital.

R8: *Passenger(?P) , Disease(?D) , hasMedicalProfile(?P,?D), hasDisease(?D,true) , DiseaseName(?D,"chronic Bronchitis"), SmartBusStation(?SBS) ,Vehcile(?V) ,connecte (?SBS, ?V),Seat(?S) , compose(?V,?S) , Actuator(?A) , Controlled(?SBS,?A) , Window(?W) , Services(?S) -> ProvideService(?SBS, ?P) , ServicesVehicle (?S,"Bus") Vehcile(?v) , takeSeat(?P, ?S) , seatType(?se, "near to window") , trigger(?A,?W) , act (?A, ?W) , openWindow(?W, true).*

The rule8 represents an example of the smart bus stop dealing with a passenger whose medical profile contains a disease. The passengers benefit from a specific service appropriate to their disease requirements. In addition, these types of people benefit from a message of notification to remember them of the time of drug, doctor suggestion, or visit the doctor.

R9: *Passenger(?P) , PersonnelProfile(?PR) , hasPersonnelProfile(?P,?PR) , Age (?A) , HhasAge(?PR,?A) , AgeValue(?A,val) , SmartBusStation(?SBS) , swrlb : greater -Than (?v , 65) , ->AlderPassenger(?P).*

R10: *Passenger(?P) , PersonnelProfile(?PR) , hasPersonnelProfile(?P,?PR) , Handicap(?hA) , AgeValue(?A,val),SmartBusStation(?SBS) , Vehcile(?V) , Seat(?S) , compose (?V , ?S) -> ProvideService(?SBS,?P) , ServicesVehicle (?S,"bus") , SeatType(?S," handicap").*

The smart bus stop also can use the personnel profile information like Age, Handicap, etc., for example, to detect the profile of passengers to direct appropriate services as in rule 9 and rule 10. The medical and personnel profile can also be used to help diagnose passengers' health.

7.6.2.4 Query Engine module

We use a query engine to handle queries received from the application layer that present a request of passengers to get a service and to get interesting information to help improve public transportation management. Some examples of questions and their corresponding SPARQL query are presented in the following:

Q1: List all passengers who request transport and their smart bus stop connected.
SELECT ?Passenger ?SmartBusStop
WHERE {?Passenger TPM:requestTransport ?SmartBusStop}
Q2: determine the number of passengers who request transport for each smart bus stop.
SELECT (count (?Passenger) as ?numberOfPassengers) (?SmartBusStop)
WHERE { ?Passenger TPM:requestTransport ?SmartBusStop }
GROUP BY ?SmartBusStop
The SPARQL query engine can access the knowledge base to retrieve patient profile information.

Q3: Identify the passengers who have a disease and benefit transport.
SELECT ?Passenger ?disease ?Vehicle
WHERE {?Passenger TPM:requestTransport ?SmartBusStop.
?Passenger TPM:benifitService ?Vehicle.
?Passenger TPM:hasMedicalProfile ?MedicalProfile.
?Disease rdfs:subClassOf MedicalProfile
?Disease TPM:diseaseName ?disease

```
}
```

Q4: display the smart bus stop and its services that are provided in last hour.

```
SELECT ?SmartBusStop ?service
where { ?SmartBusStop TPM:provideService ? passenger
?passenger TPM:benifitService ?service
?service TPM:hasTimestamp ?timestamp
FILTER (?timestamp >= ?startTime && ?timestamp <= ?endTime) }
```

Q5: Determine the passengers have benefited an emergency care service.

```
Select ? passenger ?Services
where { ?smartBusStation TPM: provideService ?passenger.
?Services TPM:hasType ?ServicesVehicle
Filter(?ServicesVehicle=”emergency car”) }
```

Q6: Display the average of passenger number by bus?

```
SELECT (AVG(?passengerNumber) AS ?averagePassengerNumber)
WHERE { ?bus :hasPassengerNumber ?passengerNumber . }
GROUP BY ?bus
```

7.6.2.5 Services and Applications Layer

Provide passengers with an interface to help them access and request the system's services. It accompanies them in getting the appropriate service for their health situation. This layer allows notifying passengers of the results of diagnostic and service offerings. It can automatically send an alert to hospitals, or other organizations can intervene to protect the passenger if a health situation is in danger. The notification includes a message that can categorize their destination:

- Message to passenger: stay safe at home, you need a doctor; you need to take your drug. If you forgot your drug check at a nearby pharmacy, your situation requires a hospital; please respect the instruction, etc.
- Message to the driver: Run slowly; Run speedily; Change the line to, etc.
- Message to hospital: In emergency cases, check patients' medical profile; We need more information about the passenger. This case needs to verify the result of the diagnostic.

The system enables communication, handles the issue of knowledge transfer across heterogeneous IoT devices and between two domains, and uses the Semantic Web to infer new knowledge. However, it exploits additional data that increases its processing time, which it manages to control through the proposed IoT architecture.

7.6.3 Use cases of the Proposed system

As many users can use the proposed system, such as passengers, doctors, hospitals, transportation management, and researchers, it can also use for many functionalities:

- Enables early diagnosis of disease by treating the data collected from passengers,
- Enables tracking the passengers, especially those with chronic diseases outside the home and hospitals.
- Collaborate with the health actors like doctors and hospitals to provide more protection for passengers through exchanging data and recommendations.
- Provide a great source of information for transportation agencies to detect overloading, improve management, and enhance service quality.
- Services recommendation: based on the data processing results, providing a recommendation represented by a set of services or suggestions for avoiding some activities, foods help passengers in daily life.
- Enable disease detection through analysis of the collected data, historical diagnosis, medical and personal profile, and other information stored in different levels of approach. In addition, using different capacities, IoT technologies, besides all tools, ease the analysis, calculation, and decision-making.

7.6.3.1 A scenario of a passenger using the proposed system

This section describes how the approach works. A use case scenario is represented by a person with a wearable device enabling him/her to make health measurements and arrive at a smart bus stop to get transport. Through the address, MAC of his/her wearable device, or his/her smartphone, the smart bus stop detects this address and sends a message asking him/her to enter his/her destination. After assigning its destination, the bus stop considers this person as a passenger *<MTP: Person MTP: requestTransport MTP: SmartBusStop>*.

To respond to the request, the smart bus stop checks the nearby vehicles' state *< MTP: smart bus stop MTP:collect-Data-From MTP: vehicle>* to identify the number of seats available. It also starts getting and gathering data transferred from the wearable devices of this passenger about his/her health status *< MTP: wearable MTP : makeMeasurement MTP : passenger>* and his/her medical profile that is stored in their wearable device in addition to a history of travels.

According to the size of the data, the medical profile information and others were stored in a wearable device, a smart bus stop, a small bus station, or in a transport agency. In the case that does not have this information or in a wearable device, the smart bus stop can connect to a small bus station that manages this smart bus stop. The small bus station provides a smart bus stop with needed information like the history of the travels of passengers. If the small bus station also does not contain passenger data, it connects to the transport agency to retrieve it. As the transport agency has high storage capacity, it provides a small bus station with all the management and services information. In addition, the transport agency can connect to the hospital to exchange health status information.

After receiving the required information, the smart bus stop, based on its system that includes different modules like ontology, SWRL rules, inference engine, and SPARQL queries, provides a representation of the content of these resources. It interprets it semantically to provide suitable recommendations represented by a set of services *<MTP: smart-bus-stop MTP: provide-Service MTP: Passenger>* and/or even alerts in risky cases. According to the results of the diagnosis of the health status of passengers, the smart bus stop provides the passenger with services appropriate to their health status. If the health status of the passenger is normal, the smart bus stop offers the next empty vehicle without other recommendations. Nevertheless, if the passenger's health status is abnormal, for example, has a fever, little difficulty in respiratory, blood pressure stage1, etc.; the bus stop provides her/him by the next empty vehicle with some recommendation like an open window, switch on the air-conditioned, etc. It sends a message to inform her/him about their health status, remember he/she to get their drug or check their doctor. The smart bus stop can provide him/her with a small car whose driver has taken health precautions where he has a contagious disease. In case the health status of the passenger is risky, the smart bus stop that infers this result provides a rescue car to travel them to the hospital. It also sends an alert to the hospital or its doctor by a message containing the health status of the passenger.

Example: Maria in Figure 7.6 is a passenger who wears a smartwatch; she can take measurements and exchange health data with her smartphone. She logged in at the smart bus stop and entered her destination after receiving a message asking her about the destinations. The measurements and medical profile data were transferred as follows: Blood Pressure =138mmHg, Heart Beat = 105, Body Temperature=38 C, and personal information: Age: 48 years, Gender; female. After collecting data about the vehicle, the smart bus stop system loads data about passengers using ontology and applies the SWRL rule 4 (see section 7.6.2.3).

....Measurement(?M1), Property(?Pr1) , propertyName(?Pr1, "bodyTemperature") , re-

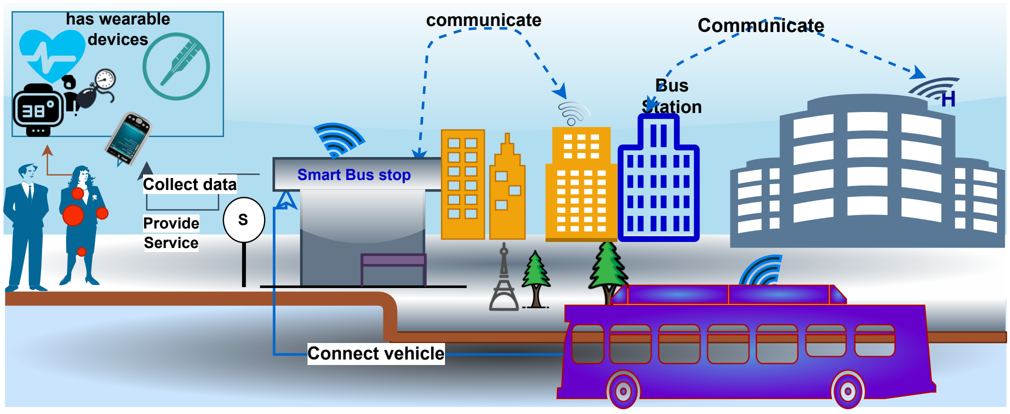

Figure 7.6: Scenario of a passenger with the proposed system [13]

latesToProperty(?M1,?Pr1) , hasValue(?M1,?hv1) , swrlb : greater Than(?hv , 37) , swrlb:lessThan(?hv, 39), Measurement(?M2) , Property(?Pr2), propertyName (?Pr2 ," Blood sugar ") , relatesToProperty(?M2,?Pr1) , hasValue(?M2 , ?hv2) , swrlb : greater Than (?hv2, 140), swrlb:lessThan(?hv2, 199) Measurement(?M3) , ,Property(?Pr2), propertyName(?Pr3 , " Blood Pressure ") , relatesToProperty(?M3,?Pr3) , hasValue (?M3 , ?hv3) , hasValue (?M3 , ?hv4) , swrlb:lessThan(?hv3 , 139) , swrlb:greaterThan(?hv3 , 130), swrlb:lessThan(?hv4 , 89) , swrlb:greaterThan(?hv4, 80)-> provideServices (?SBS , ?P) , VehicleType(?V) , takeSeat(?P, ?se) , seatType(?se, "near to window") , Arc-Switch/on(?Ac, true) ,send Alert(?SBS , ?P),alertMessage("you have hypertension stage1 , prediabets please do an examination")

Based on the measurement value, the system provides Maria with a seat on the bus and sends her a message about her health care, having her drugs in time, and avoiding any physical work. The results are stored as the history of travels of Maria passenger. Applying the proposed semantic web framework in public transportation and healthcare enables handling the heterogeneity of data and achieves alignment across two domains. However, the evaluation of the proposed framework has not tackled the vagueness and imprecise IoT data, so it needs another application where the accuracy of the result of treatment is critical, and the error is unacceptable.

7.7 Application of the proposed semantic web framework in health monitoring of COVID-19 patients

In several sectors, the integrated Internet of Things aims to improve efficiency, safety, sustainability, and quality of life. It offers many capabilities to enhance numerous sectors, thereby improving daily life and the economy. In many areas, especially those involving critical decisions or major consequences, the precision and accuracy of results are important. To maintain patient safety in healthcare, accuracy in diagnosis and treatment is crucial. An incorrect diagnosis or treatment can severely affect patients' lives and health. Particularly when dealing with viruses. Physicians need an accurate diagnosis to identify infected patients to prevent the spread of virus and to offer effective treatment.

7.7.1 Coronavirus pandemic

COVID-19 is a virus that attacks the respiratory system of persons due to an extremely serious pandemic in the world. The virus is very dangerous due to the death of millions of people. Specifically, its most recent versions are distinguished by their infectious rate. Following a challenging experience with quarantine, researchers advise using a specific procedure to combat this infection. They advise focusing on social isolation, wearing the mask, early identification of a large number of people, providing appropriate and timely treatment for those who have been infected, providing the necessary resources for treatment, and making sure hospital beds are available in case of emergency. At last, they encourage vaccination. Another solution used to front the pandemic is technology, which offers a variety of online activities like education, employment, and other things. The Internet of Things is a technology that can also aid in protecting persons from the virus. Due to its accessibility, quick user responses, capacity to provide crucial information, and ability to promote contact with others without requiring human interaction, many experts advocate using this technology [205]. The Internet of Things allows the collecting, processing, and exchanging of data about the COVID-19 that improves protecting, diagnosing, and monitoring the patients. Vital signs are data that can be used to detect any change in the patient, especially those infected by the virus. Doctors and nurses use vital signs to make decisions, whereas the internet of things can use them for the same things. The Internet of Things can measure and exchange these data to diagnose and report on patients' health. In order to effectively front the rapid spread of coronavirus, it is crucial that the internet of things responds quickly and utilizes all available resources. It must also address challenges posed by heterogeneity, vagueness, and imprecise data to achieve accurate and precise decisions. As the proposed semantic web of things framework enables to handle these issues, we use it to determine the infected patient and to provide continuously monitoring of COVID-19 patients, then provide them with different services according to their health status. Since the proposed framework is based on IoT architecture, the latter needs modeling similar to a real case that succeeds to front of the virus by providing the right services at the appropriate time and with the right resources.

7.7.2 Protection policy in Algeria against COVID-19

The health sector in Algeria, which comprises multiple clinics outfitted with various medical tools and overseen by physicians, nurses, and administrators, is dealing with the COVID-19 epidemic [206]. It employs a method that divides Algerian medical facilities into three categories: hospitals, local public health establishments (LPHE), and centers of treatment. Each class is assigned particular duties that need certain personnel and material resources, capabilities suited to the population size, and services that must be provided. A general practitioner and a nurse run the small building that serves as the treatment center. It offers basic healthcare services to the locals that live on the same street. The LPHE is bigger and has more physicians and nurses with specialized training, but it still requires beds, emergency care, and some medical procedures. The hospital is a big medical facility where thousands of nurses and tens of physicians work together to treat various life-threatening conditions. Patients with COVID-19 are given a specialized diagnosis from the hospital, including medications, respiratory supplies, etc.

7.7.3 Modeling COVID-19 patient health monitoring using the proposed framework

Fronting COVID-19 requires a set of ways, including infection prevention, detection, diagnosis, and treatment measures. The internet of things through the health monitoring of people can assist in realizing these actions. On the other hand, each health status of a COVID-19 patient needs a specific diagnosis and treatment at the appropriate time. Using an IoT architecture of the proposed framework allows the distribution of the processing and eases the resource's management. As the IoT architecture consists of a set of layers, we have inspired the tasks of each layer from the health sector policy of Algeria for facing COVID-19, which is shown in Figure 7.7.

7.7.3.1 IoT device level

A group of small and inexpensive sensors is included in the lowest level of the architecture. It can integrate into wearable devices to make it possible to collect and transmit patient vital signs and other data. Vital-signal data can be used to detect changes in patient physiological parameters, which can help healthcare personnel make clinical decisions about the treatment of patients with COVID-19. After vital-signal sensors collect data, it needs to be processed and stored in an efficient manner to allow for interpretation and management of the health status of the patient.

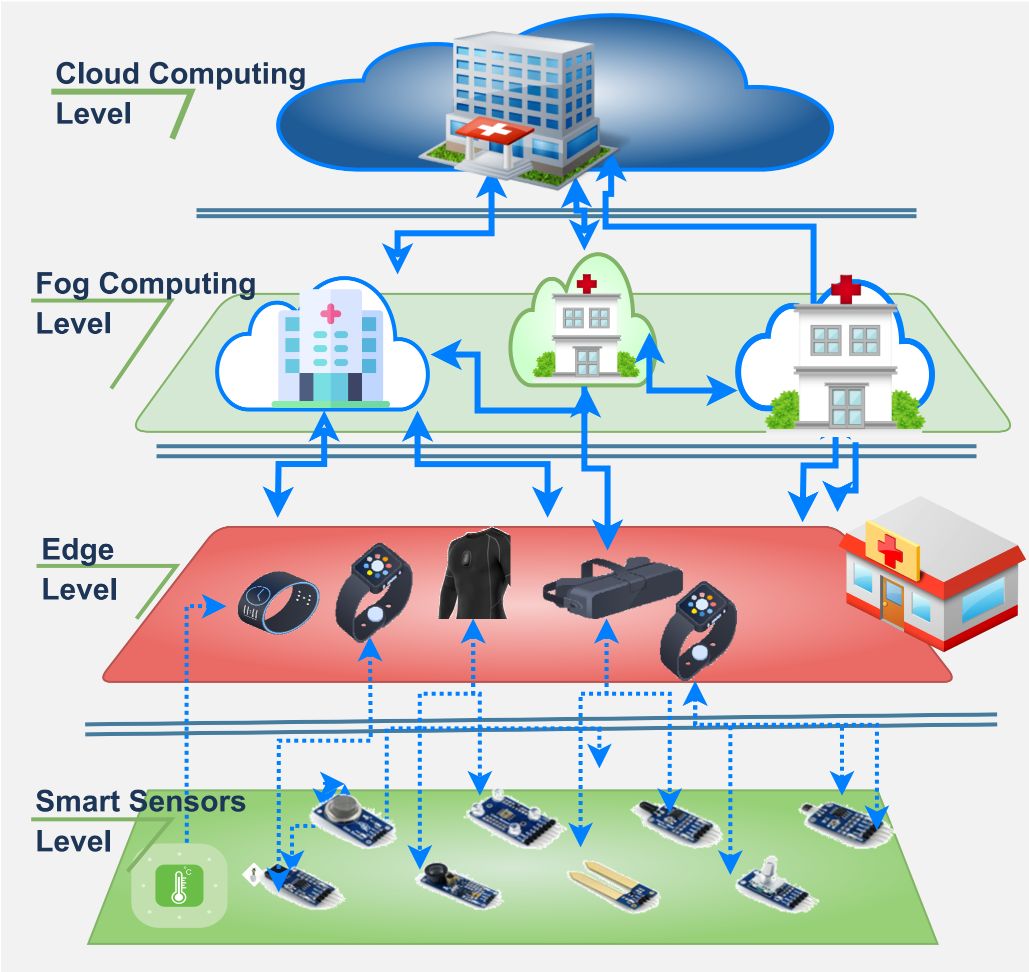

Figure 7.7: Proposed IoT architecture

7.7.3.2 Edge level (Smart devices)

The level represents wearable or portable smart items that can process and store data, such as smartphones, smart wearables, and smart sticks. These smart devices carry out precise activities because they suffer from limited processing power and memory storage. This layer is similar to the treatment center that offers basic services close to the patient. It is charged by gathering raw data from intelligent sensors, processing it, and then presenting the result to the patient or sharing it with high-level architecture. It provides real-time processing, and that ensures monitoring of COVID-19 patients since it is a near-person sensor. Due to its restricted processing and storage capabilities, the smart edge level can't always give the necessary diagnostic and deliver the right outcomes.

7.7.3.3 Fog level

The level is equivalent to local public health establishments (LPHE), which offer various services to COVID-19 patients according to their capacities. The layer employs a number of nodes to facilitate data processing and sharing. The fog node is a medium technology that is provided to get over slow networks when sending data to the cloud. It can take the role of the cloud computing infrastructure thanks to its processing and storage capabilities. The main objective is providing patients with information on current diagnoses, doctor recommendations, and test findings, and enabling the interchange of this information, particularly in the case of infected individuals, guarantees that patients receive accurate diagnoses. The fog nodes can be distributed in places shared by a large number of people. It also can arrange through its capabilities and coverage zone for delivering a proper diagnostic through its data or delegating work to those more competent, just as the tasks of the public health establishment. The fog uses the gathered information, particularly the COVID-19 findings from smart devices and the existing contact, to discover new probable infected cases since a large number of people share the same location where the infection increases.

7.7.3.4 The cloud level

The top level is similar to a hospital that optimizes handling a large number of patients and uses large amounts of data to provide precise diagnoses of patients. The cloud level is represented by servers that can process and then finally store a large amount of data. The cloud level uses IoT data obtained from sensors for a deeper diagnosis and stores data about the patient's history, diagnosis, medical history, test results, and other information in various formats, including text, image, video, etc. It also helps to provide data that can be analyzed to determine the overall pandemic situation to follow an appropriate strategy. Researchers and hospitals can use the data that is stored in the cloud in the future.

7.7.4 Modelling of health monitoring COVID-19 patients using the proposed framework

In the first application, the semantic web framework uses the acquiring data layer, semantic layer, and application layer to provide users with the result represented by different services. To improve the results offered by this framework, we need to use the fuzzy layer to process the vagueness of IoT data. The fuzzy layer consists of modules that integrate with the semantic layer modules, in which every module produces an output that another module uses as input. In the following, we detail the essential modules that contribute to output framework improvement.

7.7.4.1 Fuzzy ontological module

The fuzzy ontology module aims to make it possible to express and share knowledge that might be vague and imprecise. The fuzzy ontology represent an extension of the classical ontology. The standard SAREF ontology is classical ontology has reused to represent IoT data in the proposed framework. The ontology is characterized by its modularity, extensibility and alignment support, which facilitates its enrichment to represent such a domain and different applications.

As SAREF ontology is defined for providing a formal representation to IoT domain, the ontology is missing the concepts associated to monitoring COVID-19 patients. Furthermore, the ontology has a limit to tackle the vagueness of concepts. To define an quality ontology by avoiding of designing a specify ontology has sharing and alignment problems, we need a standard technique to create a fuzzy ontology from the classical ontology. The IKARUS-onto is methodology consists of a number of steps, beginning from the acquisition of the classical ontology and ending with the delivery of a high-quality fuzzy ontology.

For defining a fuzzy ontology, we follow two methodologies. Firstly, we enrich the extension of SAREF(SAREF4wear) ontology to represent the monitoring of COVID-19 patients by following scenario 8 of Neon methodology (see section3.4.7.2). Secondly, We also employ the IKARUS-Onto methodology to create a fuzzy ontology from a SAREF that represent the monitoring of COVID-19 patient [14]. Finally, the suggested Health Monitoring COVID-19 Patient's Fuzzy Ontology (HCFo) will be implemented and validated using the protégé 4.3 tools.

7.7.4.1.1 Enrichment of SAREF ontology to monitor COVID-19 patients' health

Enabling the ontology of the proposed framework(SAREF) provides a formal representation offer to monitor COVID-19 patients' health needs a framework that enables the reusability and enrichment of the existing ontological resource. The Neon methodology enables the construct of ontology by providing nine scenarios(see section 3.4.7.2) that encourage the creation of coherent, modular, consistent, and reusable ontologies. Saref4wear an extension of SAREF ontology aims to support the expression of the wearable device, which includes a set of sensors for data collection. However, it occurs limitations when we represent COVID-19 patients' health monitoring. It does not answer all the domain requirements, particularly regarding COVID-19, its symptoms, diagnoses, patient profile, etc.

Because the results of scenario 3 are not fully useful in their current form, which necessitates reorganizing ontological resources to achieve the desired output. As the SAREF4wear supports the extension, we apply scenario 8 to extend the ontology by adding new concepts and relationships. The following questions enable us to define the necessary concepts to represent the health monitoring of COVID-19 patients.

- What are the symptoms of Covid-19 disease?
- What treatment should be selected?
- What are the treatment limitations?
- What contraindications does the drug have?
- What should lab tests order?
- State your personal information.
- What are the parties that should intervene?
- When one or more vital signs change, will the diagnosis or treatment change?
- When establishing antecedents, is it associated with comorbidities?
- Determine the type of treatment preferred for a particular disease.
- List the diseases you suffer from.

Through the previous question's response, we define new concepts (classes), the main ones we explain in Table 7.5.

Ontology comprises two types of properties: object properties, which reflect the connections between classes, and datatype properties, which show the link between a class instance (i.e., a person) and its value. Some object attributes and datatype with their domain and range are shown in Table 7.6.

These adjustments to the SAREF4wear ontology result are shown in Figure 7.8. The ontology's classes, attributes, and instance (class members) enable the representation of data from COVID-19 patients' IoT-based health monitoring and make it easier to share the produced data.

Class	**Explain**
Patient	Represents a person who has a problem with his health and uses wearable devices to monitor them.
MedicalProfile	Represent medical data that help in the diagnosis. It covers instances like BloodType, Smoking, Alcohol intake,drug abuse, and chronic disease, Medication.
PersonalProfile	Describe the personal information. It uses in detecting andtracking the patient. It covers instances such as age, Height, weight, Activity, etc.
HealthStatus	Represent the health of patients after diagnosis.
CovidTest	Describe the test for detecting COVID-19.
Vaccinations	Represent the state of the patient with the vaccination
DiseaseHistory	Covers chronic disease, old infections with COVID-19
Hospital, Doctor,	Represent actors contribute to protecting patients
COVID-19	Describe the panademic covid-19.
Vital signs	An instance of class Property
SPo2, Pulse, Blood-Pressure, oxygen saturation	Represents the vital signs: oxygen saturation, Body Temperature, Blood Pressure, Heart Rate
HistoryDiagnosis	Represent the history of patients, especially with COVID-19.
BMI	Describe the Body Mass Index of the patient.
Message	Text exchange between system and user or others.
Symptoms	Represents covid-19 symptoms.
fever, short breath, cough	Somme symptoms has occurs when the patient infected by covid-19 virus.
Infection	Represent the health describe the Body Mass Index of patient state of patients with COVID-19.

Table 7.5: Lists the classes that be added to the standard SAREF4wear ontology

Relation	**Domain**	**Range**
hasSymptom	Patient	Symptoms
hasHealthStatus	Patient	HealthStatus
SendMessage	Smart devices	Ambulance, hospital; doctors
Recommendtest	Smart devices	Patient
ConfirmInfection	Test	Patient
hasMedicalProfile	Patient	ProfileMedical
hasSmoking	Patient	Smoking
hasPersonnelProfile	Patient	PersonnelProfile
provideServices	WearbleDevices	Services
CommunicateWith	WearbleDevices	Hospital, ambulance, fog, cloud
hasAge	Age	Integer
hasValue	Measurement	String
HasMeasurmentUnit	Measurement	String
HasMeasurmentTime	Measurement	DateTime
DiseaseName	Disease	String
TestResultValue	TestResult	Boolean
MessageText	Message	String
VaccinationValue	Vaccination	Boolean

Table 7.6: Lists of some object properties and data type defined to enrich the ontology

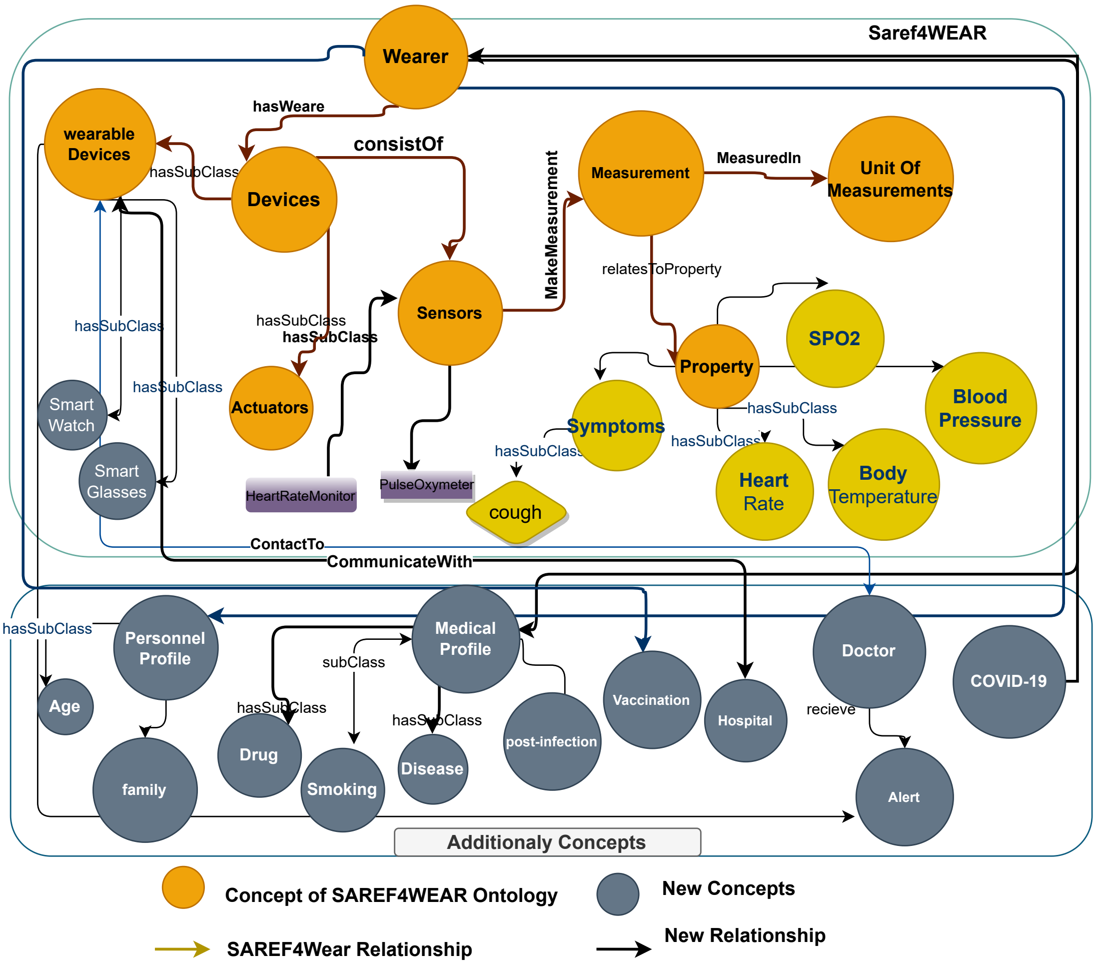

Figure 7.8: Classical Ontological model for IoT-based COVID-19 health monitoring.

7.7.4.1.2 Fuzzy ontology of COVID-19 patients' health monitoring

By offering an explicit and formal description of knowledge, ontologies are crucial for promoting domain-wide sharing and reuse. Nevertheless, classical ontology models might be unable to manage some of the knowledge related to the healthcare domain, since it is vague or imprecise. Researchers have determined that creating a fuzzy ontology is the most effective solution to this challenge [128]. Converting a classical ontology to a fuzzy ontology presents a shortcut to optimizing time, cost, and effort. The IKARUS-Onto methodology is a unique approach that expands classical ontologies to fuzzy ontologies [10]. This methodology ensures that the creation process of fuzzy ontologies is correct, shareable, and reusable. The IKARUS-Onto methodology consists of five steps: (1) the acquisition of the classical ontology, (2) the presentation of the fuzzy need, (3) the definition of the fuzzy elements, (4) formulation, and (5) the validation of the fuzzy ontology. To convert our classical ontology, we follow this process:

A) Acquiring the crisp ontology

In this process, the starting point of the fuzzy ontology development is the acquisition of the ontology. The previous COVID-19 health monitoring ontology is the classical ontology that will be used for developing a fuzzy ontology. Classical ontology has difficulty in describing vague and imprecise knowledge.

B)Presenting the need for fuzziness

The proposed ontology represents health monitoring characterized by vagueness within the domain, especially in vital signs measurement. The previous ontology needed to be fuzzy by solving the issue concepts, including some cases of vagueness that need to be determined and tackled. For example, Age (because some persons are borderline age), hasBloodPressure(because some Patients are borderline blood pressure value).

C)Definition of the fuzzy elements

According to El-Sappagh et al. [207], to define these elements, there exist two methods, The first one is to view each numerical feature as a concept and then define a fuzzy sub-concept for each linguistic value. The second method is based on defining the fuzzy datatype and fuzzy concrete role (fuzzy datatype property). The authors declared that the first method complicated the outcome and the fuzzy semantic retrieval algorithm. In our case, we firstly present fuzzy elements in our ontology, then we define the fuzzy datatype. The fuzzy elements consist of the fuzzy concepts and fuzzy relations represented.The fuzzy concept is the main component of fuzzy ontology. To some degree, it allows specifying and expressing concepts (classes) that belong to another class (superclass). For instance, we take the fuzzy concept *OldPatient* which represents an elderly person who suffers significantly from COVID-19 than other people. *OldPatient* is a sick person has age between 60 and 72 years. Using the *OldPatient* concept in ontology is a vague meaning and its instance like *Patient1* who has 69 years can be expressed with a membership degree equal (0.7).

The fuzzy object property and the fuzzy datatype property are the two sorts of relations of fuzzy ontology. These relationships between ontology entities link more than one set of things. The fuzzy object property enables the association of two instances with varying degrees of strength. For example, *"patient01 hasInfectdBy COVID-19"* with a particular degree equal to 0.8. and "patient02 hasInfectdBy COVID-19" with a particular degree equal to 0.5. The degree of vagueness of the two examples represents the size of the infection and can be determined by the doctor using the results of laboratory analysis of the infected patient's blood. The fuzzy object property also enables the association of one concept with other concepts with the same relation and different degrees. For example, *"patient01 hashealthstatus Risky"* with a particular degree equal to 0.6, and *"patient01 hashealthstatus Abnormal"* with a degree equal to 0.4.

The second kind is a fuzzy datatype property, which associates certain individuals with fuzzy data types. It connects a class instance to a literal one at a certain level. It is used for supporting the representation of crisp numerical values with linguistic values. For instance, the patient has 66 years represented with relation: *"Patient1 hasAge old"* at degree 0.5 where the Data property "hasAge" links instances "Patient" and fuzzy data types "old" with a degree.

Among the crucial concepts in our ontology, we find the *vitalsigns* concept. It includes body temperature, SPO2, respiratory rate, oxygen saturation, and blood pressure concepts. These concepts are distinguished by their nature, which involves borderline cases. To represent instances (individual) of these concepts, we need to fuzzify them. Usually, fuzzy degrees are assigned by domain experts based on degree interpretations. On the other hand, defining vital signs ranges is a universal challenge. It can be challenging to define fuzzy membership functions with high confidence. We employ the national early score 2 (NEWS2), a standardized clinical scoring system that is regarded as a good approach for inferring COVID-19, to get around this problem [208]. The system provides scores "3,2,1, 0,1,2,3" for each vital sign, which represents ranges of values. The risk was increased by adding the value "1" to the score of "0," which denotes a normal state. The definition of fuzzy sets for each fuzzy variable is made simpler by converting these scores to linguistic terms. For example, "0" presents normal, "1" presents "high or low", "2" presents "veryhigh or verylow" and "3" presents "Tooveryhigh or Tooverylow". However, the linguistic terms have same score are separate by their ranges. value. Figure 7.9

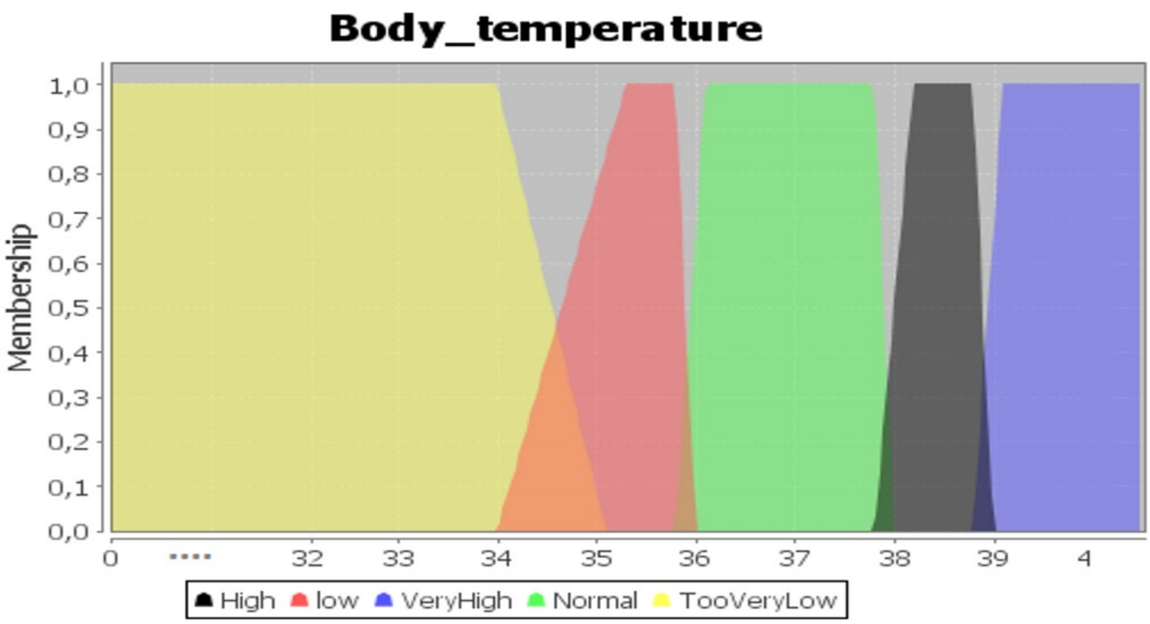

Figure 7.9: Fuzzy sets and Membership Functions for Body temperature.

presents an example of how we define fuzzy set *(TooVeryLowBodyTemp, LowBodyTemp, NormalBodyTemp, HighBodyTemp, VeryHighBodyTemp)* of the term the *BodyTemperature* by using equations of trapezoidal membership function, left-shoulder function, and right-shoulder function and on basis of the values of a and b, the numeric value has been fuzzified as follows:

$$FTooVeryLowBodyTemperature = \begin{cases} 0 & \text{x} \quad \leq 35 \\ \frac{x-34}{40-35} & b \leq x \leq c \\ 1 & x \geq 34 \end{cases}$$

$$FLowBodyTemperature = \begin{cases} 0 & \text{if } x \leq 34 \\ \frac{x-34}{35-34} & \text{if } 34 \leq x \leq 35 \\ 1 & \text{if } 35.3 \leq x \leq 35.8 \\ \frac{x-35.8}{35.8-36} & \text{if } 35.8 \leq x \leq 36 \end{cases}$$

$$FNormalBodyTemperature = \begin{cases} 0 & \text{x} \qquad \leq 35.8 \\ \frac{x-37.5}{38.8-37.8} & 37.8 \leq x \leq 38.8 \\ 1 & 36.1 \leq X \leq 37.5 \\ \frac{x-37.5}{37.8-37.5} & 37.5 \leq x \leq 37.8 \end{cases}$$

$$FHighlBodyTemperature = \begin{cases} 0 & \text{x} \qquad \leq 37.5 \\ \frac{x-37.5}{37.5-37.8} & 37.5 \leq x \leq 37.8 \\ 1 & 37.8 \leq X \leq 38.8 \\ \frac{x-38.8}{39-38.8} & 38 \leq x \leq 38.8 \end{cases}$$

Other ontology concepts that also need fuzziness are represented by the instances of the super-classes *MedicalProfile* and *PersonnelProfile*. These cases may include attribute values that take vague values, such as the concept of *Age*, which we describe with the fuzzy set *(VeryYoung, Young, Old, VeryOld)*. Smoking also is a fuzzy concept and can be defined with the fuzzy set *(little, average, high)* and reflects the daily cigarette consumption. The concept *"BMI"* refers to the definition of the body mass index (BMI), which is used by doctors in conjunction with clinical data to provide a diagnosis and also represents a fuzzy concept, which we defined by a fuzzy set *(under-weight, healthy-weight, over-weight, obese)*. The concept of *symptoms* includes all instances of typical symptoms experienced by COVID-19 patients, including chest pain, cough, diarrhea, and sore throat. These instances also have a vague meaning and need to define each of them. For example, in the instance of the symptom *"Cough"*, we create its fuzzy term *(short, moderate, long)* using a membership function based on its number in a minute.

The *healthStatus* concept represents the patient's current health status and is considered a crucial fuzzy concept in our ontology. To represent this concept correctly, we define three fuzzy sets *(normal, abnormal, and risky)*.

In Table 7.7, linguistic variables of the vital indicators, symptoms, personal information, and health status with their membership function were represented.

After defining the fuzzy datatype, the definitions of the fuzzy datatype property (concrete role) become easier. It will be developed to link each numerical value to its corresponding linguistic value. For example of the BodyTemperature linguistic variable has five fuzzy datatypes of : *TooVeryLowBodyTempearture, LowBodyTemperature, NormalBodyTemperature, HighBodyTemperature, as well as VeryHighBodyTemperature*. Using these fuzzy data types, we create the fuzzy datatype property as following: *hasTooVeryLowBodyTempearture, hasLowBodyTemperature, hasNormalBodyTemperature, hasHighBodyTemperature, and hasVeryHighBodyTemperature*. In the definition, the class *BodyTemperature* is used as a domain, and the fuzzy datatype uses as a range.

D) Formalization and validation of fuzzy ontology

The fourth step of the IKARUS-Onto process has the objective to make the machine understandable to the fuzzy ontology. After the production of the fuzzy ontology, a validation process must be performed to ensure that the created ontology accurately and adequately reflects the vagueness of the domain, which represents the fifth step.

To formulate it, we need to codify this fuzzy ontology using a tool that allows defining different ranges of its terms. We also need to tool that enables the verification and validation of this ontology. The owl2plugin enables the two operations, which we can combine the formalization and validation operations into one step. The plugin was created by Bobillo and Sraccia [136] and allows giving a fuzzification process through the annotations in the crisp ontology. It enables its formulation by integrating it into the protégé 4.3 software for easing the creation, editing, and saving of the fuzzy element. It also enables defining a Fuzzy data type for each fuzzy value and supports producing the membership functions as a tool for mapping concepts with vagueness. The plugin provides the Left-shoulder (k1, k2, a, b), Trapezoidal (k1, k2, a, b, c, d), Right-shoulder (k1, k2, a, b), and Triangular (k1, k2, a, b, c) for each fuzzy value to enable us to create the fuzzy data types, where the parameter "a" denotes the lower limit and the parameter "b" denotes the upper limit. K1 and K2's parameters denote the minimum and maximum inclusive values, respectively. The plugin provides for any fuzzy data type's range (K1, K2) and linguistic variables an annotation in order to define it. Figure 7.10 shows the outcome of developing a fuzzy ontology for COVID-19 health monitoring using the earlier software, with a sample of how to design a datatype and its notation.

After the creation of fuzzy ontology with protégé, we used it for the validation of ontology. The validation of fuzzy ontology consists of the verification of the accuracy, com-

Lingu -istic terms	**Fuzzy Linguistic terms**	**Member -ship function range**	**fuzzy concrete role**
Body-Temper -ature	TooVeryLowBodyTemp, LowBodyTemp, NormalBodyTemp, HighBodyTemp, VeryHighBodyTemp	[<35] [35.1, 36] [36.1, 38.0] [38.1, 39] [>39.1]	HasTooVeryLowBodyTemp HasLowBodyTemp HasNormalBodyTemp, HasHighBodyTemp, HasVeryHighBodyTemp
Respir -atory-Rate	TooVeryLowRespiratoryRate, LowRespiratoryRate, NormalRespiratoryRate, VeryHighRespiratoryRate, TooVeryHighRespiratoryRate	[<8] [9, 11] [12, 20] [21, 24] [<25]	HasTooVeryLowRespir.. HasLowRespiratoryRate, HasNormalRespiratoryRate, HasVeryHighRespiratory.. HasTooVeryHighRespirat..
Blood-Pre-ssure	TooVeryLowBloodPressure, VeryLowBloodPressure, LowBloodPressure, NormalBloodPressure, TooVeryHighBloodPressure	[<91] [91, 100] [101, 110] [111, 129] [<220]	HasTooVeryLowBloodPr.. HasVeryLowBloodPressure HasLowBloodPressure, HasNormalBloodPressure, HasTooVeryHighBloodPre..
Pulse	TooVeryLowPulse, LowPulse, NormalPulse, HighBloodPulse, VeryHighBloodPulse, TooVeryHighBloodPulse	[<40] [41, 50] [51, 90] [91, 110] [111, 130] [>131]	HasTooVeryLowPulse, HasLowPulse, HasNormalPulse, HighBloodPulse, HasVeryHighBloodPulse, HasTooVeryHighBloodPulse
Age	VeryYoung Young Mild old Very Old	[=<20] [16, 38] [35,50] [45, 60] [55, 65]	HasVeryYoung HasYoung HasMild HasOld HasVery Old
BMI	under-weight healthy-weight over-weight obese	[=<19] [19.5,24] [25,30] [>=30]	HasUnder-weigh HasHealthy-weight HasOver-weigh HasObese
Alcohol	Little, moderate, much	[1,2] [2,4] [>4]	HasLittle HasModerate, HasMuch
Smok-ing Cigar -ette	Light Average heavy	[>3] [3 , 10] [>=10]	HasLittle HasModerate, HasMuch
Symp -toms Cough	short moderate long	[=<5] [5,30] [>=30]	HasShort HasModerate HasLong
Health Status	Normal Abnormal Risky	[=<4] [5,6] [>=7]	HasNormal HasAbnormal HasRisky

Table 7.7: A list of some linguistic variables, linguistic terms, and MFs.

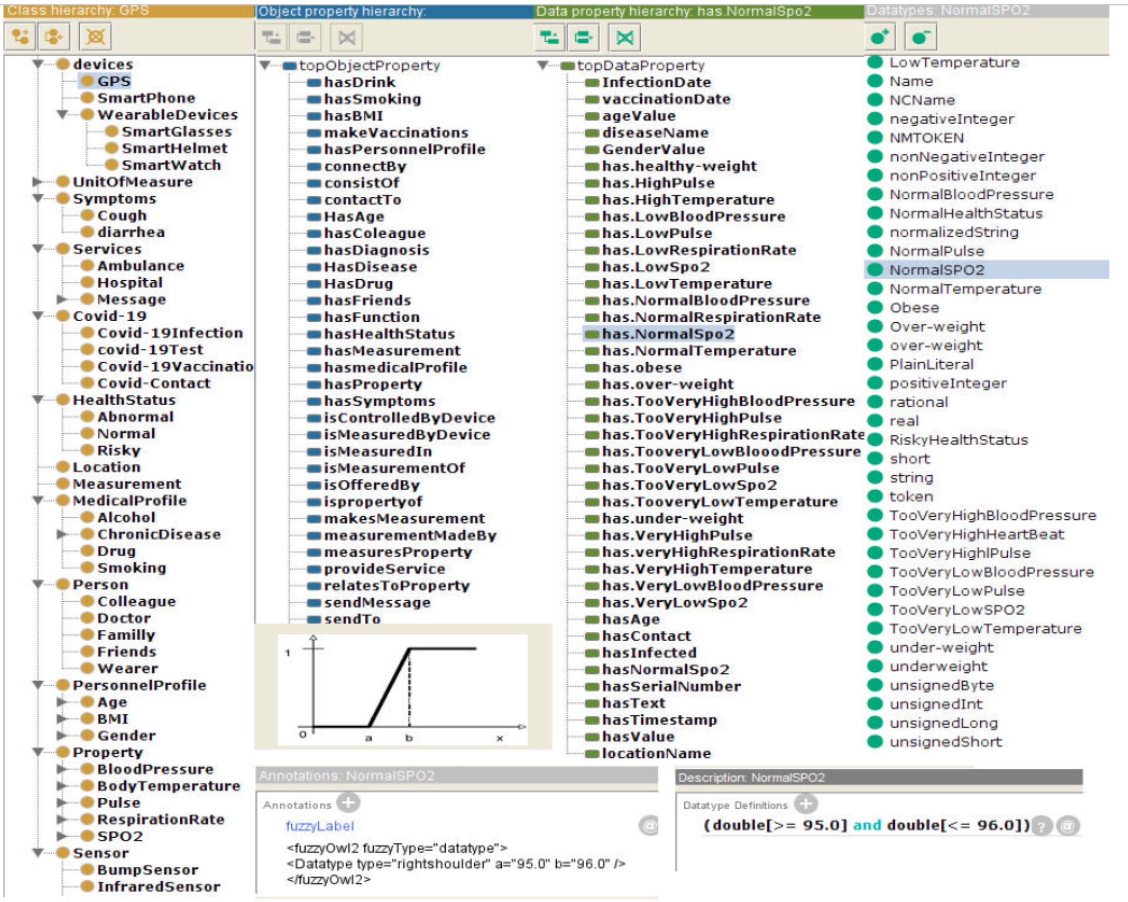

Figure 7.10: Component of Proposed fuzzy ontology health monitoring of COVID-19 patient, with samples of fuzzy datatype, annotation, and description [14]

pleteness, and correctness of the fuzzy ontology. Using Pellet Reasoner and the FuzzyDL Reasoner that is comprised in the owl2plugin, we apply these two reasoning to allow the detection of inconsistent data to correct or remove it. The other verification requirements have historically been manually completed. Using FuzzyDL, we may query the ontology and retrieve the information related to user responses. An example of a fuzzy query is the following:

Q: displays the patients' older age, low oxygen saturation, and health status.
hasAge only(Age.value only (hasValue value VeryOld)) and hasOxygenSaturation Some(OxygenSaturation.value some (hasValue value LowOxygenSaturation)) and Patient hasAge.

After the creation of the fuzzy ontology through the Ikarus methodology, the ontology will be ready to use it. As the proposed framework consists of a set of modules where ontology plays a crucial role, we will present a scenario that represents how the proposed system works.

7.7.4.2 The fuzzification module

The fuzzification is an important phase in the processing of fuzzy data in the fuzzy layer. It consists of converting the numerical data of vital signs such as oxygen saturation, heart rate, blood pressure, etc. into fuzzy linguistic variables, which can then be used in the other modules. Fuzzification is implemented using membership functions. They are also used NEWS-2 to define the relation between a numerical value and a fuzzy linguistic variable.

7.7.4.3 The fuzzy inference module

Fuzzy inference is concerned with the application of rules(if-then rules) that describe the relationships between fuzzy linguistic variables and decision-making. It is based on the Mamdani engine [130], a type of reasoning apparatus that enables constructing the relationship between data input and a plausible output, is used to depict it. This engine's structure is straightforward and supports multiple-input-single-output and multiple-input-multiple-output systems. It is suited to make a decision about the health status of patients.

7.7.4.4 The Defuzzification module

In order to use the fuzzy outputs in the real world, it is frequently essential to transform them into numerical values or binary decisions after applying the fuzzy control system rules and receiving the fuzzy outputs. Defuzzification is responsible for transforming the fuzzy output into a crisp value. Defuzzification module is based in the center of gravity method, which consists in computing the center of gravity of the weighted membership function of the fuzzy outputs [203].

7.7.4.5 Query engine module

Query engine module enables users to ask the system and get the result or retrieve information. It uses fuzzy ontology to query knowledge bases or information systems. It enables to provide of a query using SPARQL as follows:

Q1: Display people with their addresses and gender who have ages between 50 and 70 and have not been vaccinated and have test Result Positive.

SELECT ?person , ?Address? vaccination ?Age
WHERE { ?person HMC:hasVaccinate ?vaccination.
?person HMC:hasAdress ?Adress.
?person HMC:hasAges ?Age.

?TestResult HMC:TestResultValue ?test
FILTER((?age ≥ 60 && ?age ≤ 75)&&(?vaccinate! = true) && (?test = true))}

Q2: Finding the patients' health status is critical.
SELECT ?Person, ?HealthStatus
WHERE{ ?Person HMC: HasMeasurement measurement.
Measurment HMC:relatetopreperty ?BP.
Measurment HMC:relatetopreperty ?Spo2.
Measurment HMC:relatetopreperty ?BodyTemeprature.
FILTER ((?BloodP ≥ 140) (?Spo2 ≥ 140) (?BodyTemeprature ≤ 100))}.

Q3: Display the patients have COVID-19 positive cases and above age 65.
SELECT DISTINCT ?Name ?gender ?age ?Resulttest
WHERE{ ?patients HMC:MakeTest ? TestResult.
?TestResult HMC:TestResultValue ?test
?patients HMC:hasName ?Name. ?patients HC:hasAge ?age.
?patients HMC:hasGender ?gender.
FILTER (?age ≥ 60)
FILTER(?test == "true")
ORDER BY (?age)}

7.7.5 A scenario of using the proposed framework to protect people's life against pandemics

From the objective of the proposed framework is to improve the system processing to get an accurate result and provide a correct decision. Using it to safeguard people's lives against COVID-19 that is considered a reliable assessment. To do this, the suggested framework carries out many activities to process and communicate knowledge among its constituent module. The acquisition of a collection of raw data gathered by sensors that are included in smart wearable devices initiates these processing activities.

The first module that contributes to the processing of gathered data is the fuzzification module. With the use of a membership function that defines the level of fuzziness, it transforms these crisp values into fuzzy values. These data are sent together with the fuzzification process' findings to a crucial module called "Knowledge base". This module formalizes the representation of this outcome using a fuzzy ontology, personal and medical profile data, historic diagnosis, and the contacts files, for enhancing, sharing, and reuse. Additionally, the module makes use of SWRL rules, which cooperate with the inference engine to derive implicit knowledge about healthcare services or provide suggestions. Here are some examples of SWRL rules:

Rule 1: Using information collected from the sensors about the measurements of the vital signs to decide about the health status of the patient.

Patient(?P) , SmartWearableDevice(?SWD) , measureFeatures(?SWD , ?P) , Sensor(?S) , consistOf (?SWD , ?S) , Measurement(?M) , makeMeasurement(?S , ?M) , Property(?BloodPressure) , relatesToProperty(?M , ?BloodPressure) , hasValue(?M , NormalBloodPressure) , Property(?RespirationRate) , relatesToProperty(?M , ?RespirationRate), hasValue(?M , NormRespirationRate) , relatesToProperty(?M , ?NormalSPO2) , hasValue(?M,NormalSPO2) , relatesToProperty(?M , ?NormalTempB) , hasValue(?M , NormalBodyTemperature) -> hasHealthStatus(? P , Normal), messageText (?M, "protect your self from COVID-19").

If the measurement values of the patient are normal, the system concludes the health status of the patient is normal and remembers it with a message to protect himself from COVID-19. The system's services have also changed according to the health status of the patient, which is shifting between normal, abnormal, and risky.

Rule2.1: The patient has health status is abnormal.

Patient(?P) , SmartWearableDevice(?SWD) , measureFeatures(?SWD , ?P), Sensor(?S) , consistOf(?SWD,?S) , Measurement(?M), makeMeasurement(?S , ?M) , Property (?RespirationRate) , relatesToProperty(?M,?RespirationRate) , hasValue (?M , VeryHighRespirationRate) , relatesToProperty(?M , ?VeryHighTempB) , hasValue(?M , VeryHighBodyTemperature) , Doctor(?D) , Message(?Me) -> hasHealthStatus (?P , Abnormal) , provideService(?P , ?Me) , messageText (?Me , "Your health condition is abnormal, please take a COVID-19 test and take protective action to avoid contaminating the others."), messageTo (?SWD , ?D).

Providing health services that are appropriate for the wearer's health condition; for wearers who have not confirmed a positive COVID-19 result and whose health status is abnormal, the system will propose that they take the COVID-19 test and contact their doctor.

Rule2.2: The health status of the patient is risky.

Patient(?P), SmartWearableDevice(?SWD) , measureFeatures(?SWD , ?P) , Sensor(?S), consistOf(?SWD , ?S) , Measurement(?M) , makeMeasurement(?S , ?M) , relatesToProperty(?M , ?RespirationRate) , hasValue(?M, TooVeryHighRespirationRate) , Property(?Pulse) , relatesToProperty(?M , ?Pulse) , hasValue(?M , TooVeryHighPulse) , Property(?RespirationRate), relatesToProperty(?M,?BodyTemperature) , hasValue(?M , HighBodyTemperature) , Hospital(?H) , Ambulance(?A) , Message(?Me) -> hasHealthStatus(?P , Risky) , provideService(?P,?M) , contactHospital(?SWD , ?H), callAmbulance(?SWD , ?A).

Providing the patient that health status with emergency service such as an ambulance and contacting the doctor, especially when he has a test confirmed its infection by COVID-19. Rule3: Using more information than vital signs such as the medical profile, the personnel profile, and symptoms helps in determining the health status of the patient.

R3.1: Using the cough symptom in the diagnosis of the health status of the patient. *Patient(?P) , SmartWearableDevice(?SWD) , measureFeatures(?SWD , ?P) , Sensor(?S) , consistOf(?SWD,?S) , Measurement(?M) , makeMeasurement(?S , ?M), Property (?RespirationRate) , relatesToProperty(?M,?RespirationRate), hasValue(?M , NormalRespirationRate) , Property(?SPO2), relatesToProperty(?M , ?NormalSPO2) , hasValue (?M , NormalSPO2) , relatesToProperty (?M , BodyTemperature) , hasValue (?M, HighBodyTemperature) , Cough(?C) , hasSymptoms(?P,?C) , Doctor(?D) , Message(?Me) -> hasHealthStatus(?P , Abnormal) , ProvideService(?P,?Me) , messageText (?Me, ?please take COVID-19 test) , contactTo(?SWD, ?D).*

R3.2: If the patient is old age and has smoked, without any change in measurement value, the system recommends making them a test and contact the doctor.

Patient(?P), SmartWearableDevice(?SWD) , measureFeatures(?SWD , ?P) , Sensor(?S) , consistOf (?SWD , ?S) , Measurement(?M) , makeMeasurement(?S , ?M), Property(?RespirationRate) , relatesToProperty(?M , ?RespirationRate) , hasValue(?M , LowRespirationRate) , Property(?BodyTemperature) , relatesToProperty(?M , NormalBodyTemperature) , hasValue(?M , NormalBodyTemperature) , Age(?A) , HasAge(?P,Old), Smoking(?S) , hasSmoking(?P , ?S) , SmokingValue(?S , ?Moderate) Doctor(?D) , Message (?Me) -> hasHealthStatus(?P , Abnormal) , ProvideService(?P , ?Me), messageText (?Me , ?please take COVID-19 test and avoid the smoking) , contactTo(?SWD , ?D).

R3.3: If the patient has a disease like diabetes with any change in measurement value, the system recommends making them a test and contact the doctor.

Patient(?P), SmartWearableDevice(?SWD) , measureFeatures(?SWD , ?P),Sensor(?S) , consistOf (?SWD , ?S) , Measurement(?M) , makeMeasurement(?S , ?M) , Property(?RespirationRate), relatesToProperty(?M , ?RespirationRate), hasValue (?M, LowRespirationRate) , Property(?BodyTemperature) , relatesToProperty(?M , NormalBodyTemperature), hasValue(?M , NormalBodyTemperature) , Disease(?Ds) , hasDisease(?P,?Ds), DiseaseName(?S, ”Diabetic”), Doctor(?D) , Message (?Me) -> hasHealthStatus(?P, Abnormal) , ProvideService(?P, ?Me), MessageText (?Me, ?please take COVID-19 test) , contactTo (?SWD , ?D) .

Rule 4: To stop the infection cycle, the wearer who tested positive for COVID-19 must inform his or her contacts of the results.

Pateient(?P), TestCovid(?T), hasTest(?P , true), Colleage(?C) , Familly(?F) , Message(?Me)-

> sendMessage(?SWD,?C), sendMessage(? SWD, ?F), messageText (?Me, "your collegue(familly) has a positive test COVID-19, Please check your system or take a COVID-19 Test").

To protect the users from the virus and its impacts, the system offers a range of services specific to each patient's health situation. The user needs to accede to the system in order to benefit from its services. The query engine module is mostly used in these communications. Users can be patients, doctors, nurses and health administrators. they can request a different question and get appropriate answering. An illustration of a user's system y query expressed as a SPARQL.

Query1: Determine the services that provided to users' health is abnormal.
SELECT ?Patient ?service
WHERE ?Patient HCFo : HasHealthStatus ?healthStatus
Patient HCFo : BeniftServices ?Service
filter(?healthStatust = "Abnormal")

Query2: Verify which patients are most likely to be affected by COVID-19
SELECT ?Patient
WHERE ?Patient HCFo:HasAge ?Age.
?Patient HCFo: ChronicDusease ?disease.
?Age HCFo:HasValue ?V
?Patient TH:HasHealthStatus ? HealthStatus.
Filter(?V ="veryOld").
Filter(?Disease ="Diabetic").
FIlter(Result=" Abnormal")

Query3: Count the number of users has been infected by COVID-19.
SELECT (count (?Patient) as ?numberOfPateient)
WHERE ?Patient HCFo:hasTest ?Test.
?Test HCFo:hasResult ?PositiveResult.

7.7.6 Domains that can use the proposed approach to protect against COVID-19

Many sectors can use the approach for detecting patients infected by COVID-19. We list a few of these domains below:

7.7.6.1 Health domain

The health sector is the first domain concerned with facing the spreading of COVID-19 and providing the appropriate diagnosis and treatment for the patients. It can be used in the hospital to monitor the patients and informed the doctors and nurses in case of any undesirable evolution. The approach can help the sector reduce its cost.

7.7.6.2 Domain of education

Many individuals attend universities and schools. Even with the adoption of methods like remote learning and avoiding the need to go to these institutions, keeping them under quarantine for an extended period of time is very difficult. The approach enables the identification and isolation of affected individuals who directly contribute to stopping the spread of the virus through the use of health information. The wearable devices' performance can be improved by enabling the assistance of fog computing and the cloud for analyzing and processing various types of data.

7.7.6.3 Transport domain

All organs of the transport system have been influenced by COVID-19 pandemic, particularly during the lockdown. Additionally, the numerous people who utilize public transit can transfer the virus to one another while they are on the journey. The approach can use for identifying and tracking infected persons and benefit from domain infrastructure. By utilizing its resources to offer healthy mobility, public transportation contributes to the fight against the virus.

7.7.6.4 Sport domain

Generally, the transport domain depends on gatherings of people. In order to allow for an early evaluation and follow-up of infected patients to prevent the infection of other persons, the proposed approach may be simply used with this zone. Additionally, because of their enormous capacity for collecting people, sports locations can serve as a fog for wearable device data collection and enable quick treatment.

Through the proposed framework, the semantic web layer enables to tackle the heterogeneity and lack interoperability problems. Fuzzy layer also enables to handle the vagueness and imprecise of IoT data that improve the accuracy of the framework processing. However, the big size of data generated by the high number of devices that connected to framework due to large size of output. The storage and retrieve of information from this output require a new storage technique to enable the searching very quickly.

7.8 Using the storage and retrieval of the proposed framework to improve patient health monitoring

The Semantic Web of Things framework is an effective approach for addressing the challenges caused by the heterogeneity of the IoT domain and achieving semantic interoperability. In healthcare, the framework provides solutions for improving the performance and quality of healthcare services. It enables patient health monitoring, real-time diagnosis, and facilitating the management of patients, administrators, and resources. This framework leverages Semantic Web technologies such as RDF, OWL, and SPARQL. It generates RDF data as a flexible and standard method for modeling data to enable the exchange of it between different IoT applications. The RDF format allows combining data from different sources, such as hospitals, clinics and laboratories. To enable effective healthcare data management and support the development of intelligent healthcare applications, such as decision support systems, it is essential to store healthcare data in a standardized format called RDF.

7.8.1 Big RDF data storage

Heterogeneity is one of the main characteristics of data and can lead to problems in processing, exchanging, and interpreting them. To tackle this problem, Berners Lee defines the semantic web as a solution. The semantic web is the next generation of the web, intended to make the machine more understandable. It also facilitates the exchange of data between different systems and eases the search for information using their context. It uses semantic web technologies, such as RDF, which provide a common framework for describing and sharing data. RDF (Resource Description Framework) provides a flexible, extensible framework for describing data based on the concept of triplets. An RDF triplet consists of three parts: a subject, a predicate and an object. This data model is widely used to describe and share data on the Web. Many companies are using it for data representation and exchange, which is progressively increasing due to the size of RDF data. An example of the volumes of RDF data, in May 2020, there were 1,255 datasets with 16,174 links published by data sources in the Linking Open Data Project [154]. There are also many dataset someone specific to such domains (DBPedia [152], Bio2RDF [209]) that contain more than a billion triples (more 3 billion, more 2.3 billion respectively). This status led to the emergence of the term Big RDF data to express the RDF dataset that shares the volume characteristic with the big data. This term has become more appropriate with the huge expansion in the use of IoT technology, where there is an explosion of data from various sources, including sensors, mobile devices, etc. Healthcare that based on the Internet of Things is among the areas that consist of these resources and need for data integration and interoperability.

7.8.2 RDF health IoT data

The internet of things has significantly impacted and improved the healthcare domain by enabling health data collection, sharing, and processing. This has led to improved health services and reduced intervention time, especially in complicated cases. IoT integration has also allowed for remote diagnosis, which reduces the burden on doctors and eliminates the need for patients to travel to hospitals. Wearable devices like smartwatches and smart bracelets are some examples of how IoT is integrated into healthcare. The wearable devices are used to measure the patient's vital signs (blood pressure, glucose level, heart rate, etc.) and then treated or sent to doctors. Wearable devices have become increasingly important in healthcare, especially for patients with chronic diseases. To facilitate the use of wearable devices, many applications are specifically developed to monitor patients, such as BikeNet [210], sleepBpatterns [211], etc. Despite the benefits of healthcare IoT devices and applications developed using them, there is a challenge of heterogeneity and lack of interoperability due to data coming from multiple sources and arriving in different formats. This problem is due to the emergence of IoT health RDF data, which refers to health-related information collected by IoT devices and represented in RDF format. This type of data plays a vital role in improving healthcare services by providing valuable insights into patient health, medical history, treatment plans, and test results. By utilizing RDF health IoT data, healthcare professionals can make informed decisions about patient care and treatment, leading to better health outcomes. IoT health applications are mainly designed to monitor patients, and as a result, the health RDF data collected includes information about a patient's vital signs, patient information, recommendations for healthcare services, and more. Since this data is constructed based on the same ontology and SWRL rule, it is often distributed across multiple files that share the same structure. These files build a repeatable and syntactically consistent dataset. With the increasing volume of RDF health IoT data, there is an emerging demand for efficient storage and retrieval mechanisms to ensure that data is readily available in a timely manner. However, as the volume of RDF health IoT data continues to increase, there is a growing need for effective storage and retrieval mechanisms to ensure the timely availability of data. There is a need to leverage the new computing paradigms of the internet of things, which represent the real-time processing capability offered by Edge and Fog Computing.

7.8.3 Modelling of Health Monitoring using the proposed IoT architecture

IoT paradigms allow handling issues related to RDF IoT data storage, processing, and retrieving information. Cloud computing is an IoT paradigm that can deal with storing and processing large amounts of data. However, many interconnected objects connect via the internet and traverse many intermediate networks due to an overload of bandwidth and latency time. Cloud computing cannot ensure real-time processing and retrieve information very quickly. To benefit from using IoT paradigms, It is necessary to utilize them through an IoT architecture that helps to provide efficient storage and accelerate the search time. Based on the proposed IoT architecture of the proposed framework depicted in Figure 7.2, the architecture consists of a set of levels, with each level dedicated to specific tasks as follows:

The first level is connected devices or IoT sensors. It enables sensing patients or the healthcare environment and then transferring it. This level consists of millions of heterogeneous connected sensors equipped in different medical things or in smart devices worn by patients. It operates using communication technologies, such as Wi-Fi, Bluetooth, etc., to exchange IoT data.

The edge level in an IoT-based healthcare system consists of resources that are located close to the IoT sensors and are responsible for processing and analyzing the data collected from the sensors. The resources in the edge level includes edge devices such as wearable devices or smart devices that have the capabilities of temporary storage and processing using the semantic web of the heterogeneous data transferred from the lower layer (IoT sensors). The output of the layer is an RDF file containing information like the values of IoT measurements construct using ontology and other semantic web technologies.

The Fog level serves two roles: the first role is partitioning the RDF data into groups to ensure better indexing, and the second role is indexing this data. The nodes Fogs partition the data using MEWS and fuzzy logic algorithm as well as using a binary tree with two pivots to provide adequate indexing.

As the proposed IoT architecture, the top level is the cloud level, which is responsible for storing and processing large amounts of data generated by the lower layers. After classifying and indexing the data using a binary tree in the fog layer, the cloud layer store RDF data for analysis and processing, especially in complicated cases and complex treatment.

7.8.4 Modeling health monitoring with the proposed framework

Monitoring using the proposed framework is based on the semantic web for achieving semantic interoperability and handling heterogeneity data problems. It provides health IoT data with semantic processing and enables the exchange and interpretation of this data between different IoT devices. The semantic web framework benefits from the RDF model to use it to represent its output as an RDF file. As the high number of interconnected health devices is growing very quickly, the size of the SWoT framework output that is represented by RDF data also grows exponentially. To process, store, and manage these huge amounts of health data efficiently and flexibly, the proposed framework offers the clustering and indexing levels. In the following, we detail the principal layers:

7.8.4.1 Semantic layer

The semantic layer of the proposed framework is composed of several modules, each with specific tasks, and the output of one module serves as the input for the next. The semantic layer is based on the SAREF ontology, which provides a standardized way to represent IoT health data and facilitate its exchange across different applications. SAREF ontology also supports the sharing and alignment with other ontologies. The layer also includes a SWRL rules module, which consists of rules that allow for deducing and extracting hidden relationships from the SAREF ontology. Additionally, an inference engine is utilized to deduce new knowledge based on the rules defined in the SWRL module. The SPARQL engine is another essential module in the semantic component, which serves as an interface between the component and other systems or users. This module responds to user requests and enables them to retrieve information from different IoT sources or databases using the SPARQL query language. The SPARQL engine provides users with access to a wide range of data related to healthcare, such as patient data, medical records, and other relevant information.

In order to facilitate the use and exchange of data produced by the semantic layer, it is represented by RDF (Resource Description Framework) files. Each file contains information about IoT measurements, and patient data, including medical profiles, diagnosis results, and recommendations for drugs, foods, and activities. Using RDF files enables healthcare professionals to retrieve information about patients with similar vital signs, facilitating personalized medicine and tailored treatment plans based on individual vital signs and medical history. The files also provide alerts and notifications when measurements deviate from normal patterns, allowing doctors to take prompt action to address potential health problems. Due to the high volume of RDF data generated by the system, effective solutions for organizing and structuring the data are needed to minimize storage and search times. This requires advanced storage and indexing techniques that can effectively manage and retrieve the data stored in RDF format. By adopting efficient storage solutions, the healthcare domain can improve the quality and speed of patient care and reduce the likelihood of errors or delays in treatment

7.8.4.2 Improve the clustering of RDF files using fuzzy logic

Indexing structures are used to organize a database to speed up search time by reducing the amount of data that needs to be searched. Partitioning is a method that may be used to separate the data into more manageable, smaller groupings depending on certain criteria, such as a range of values or individual property. The data may be divided into manageable subsets, making creating an index for each one quicker and less expensive. Partitioning also makes it possible to direct searches directly to the relevant portion of data rather than scanning the full database, which may enhance query speed. IoT healthcare data that is collected from patients and used to calculate their health status can be divided into several categories. Physicians and nurses use tools to classify the health status of patients. The Modified Early Warning Score (MEWS) is a tool used in healthcare facilities to help determine the health status of patients. It provides a score allows classifying patients into different risk categories according to their health status. An overall MEWS score is between 0 and 15, with a score of “0” indicating that the patient’s health status is normal. As long as an increase in this score by a value of “one” means that the health status has changed, we can define classes with the following ranges: (class1 has range[0,1], class2 has range [1,2], class3 has range [2,3], class4 has range [3,4], class5 has range [4,5], class6 has range [5,6]... class15 has range [14,15]). As health IoT data measurements are distinguished by their imprecise and vagueness nature, using these data without treating this problem due to incorrect classification is unacceptable in the healthcare domain. A fuzzy logic algorithm is an extension of logic Boolean which uses the fuzzy set theory to provide a value of truth between one and zero. Using fuzzy logic, we can provide the patient’s health status with a correct classification. The clustering layer benefits from fuzzy layer modules. The fuzzy layer enables the calculation of the value of health status to improve the classification of RDF files into one of 15 classes of fuzzy systems. For example, An RDF file contains patient vital signs measurement *(blood pressure=B1, heart rate=H1, oxygen saturation=S1 body temperature=T1, respiration rate=R1)*. According to MEWS, the system converts and calculates the health status of this patient using the rule:

Rule1 *= IF ((blood pressure is Normal) and (SPO2 is Normal) and (Pulse is Normal) and (RespirationRate is Normal) and (BodyTemperature is Normal)) then (HealthStatus* *is Normal).*

As a result of the calculation where the value of health status is less than 1, the RDF file will be classified in a normal class. Despite the partitioning of RDF files in class enabling to minimize the space for searching, each class of files still also needs to be structured to accelerate the research.

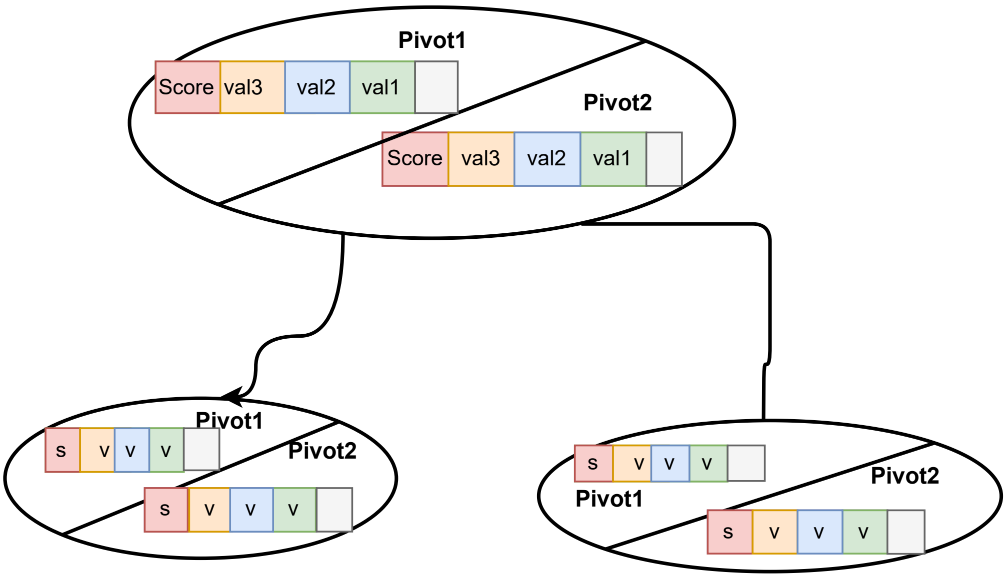

Figure 7.11: Tree with pivots structure

7.8.4.3 Indexing layer for health monitoring

Indexing techniques are used to organize and manage large databases in order to improve access times and reduce search space. Many different indexing techniques are available, and one of these is the B-tree (Binary tree), a simple tree-based structure in which nodes can have no more than two children. The tree's simple structure makes searching operation very fast, and the insertion process facilitates. The tree has two pivots in its node to improve search efficiency (see figure 7.12). The tree's construction depends on many clusters (in our case, MEWS permits defining 15 classes). For each cluster, we defined the same steps to insert an RDF file. Each file is associated with a cluster and inserted into the tree corresponding to that cluster. The global score for each file is calculated using a fuzzy algorithm to determine the appropriate node to insert the file.

The insertion process begins by adding the score value to a vector that includes measurement values, and then checking the node. If the node in the tree does not have pivots, the new vector is inserted as a pivot or used to calculate its distance from the two pivots. The distance is calculated using Euclidean distance where its equation is:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

where Xi is vital signs values and score value of pivot1 and Yi the vital signs value and score value of input vector. The distance between pivot1 and pivot2 with vector input is calculated with Euclidean equation. If the distance D1 $<$ D2 is similar to pivot 1, the vector is inserted into the left child node; otherwise, it is inserted into the right node. This process is repeated until a new node is created, and the vector is inserted as pivot 1.

Before starting the search of an RDF file using the tree, a preliminary step is to compute the score in order to identify the relevant group and the corresponding tree. After that, an input vector is created using this score and the vital signs. The search is based on calculating the Euclidean distance between the input vector and the node pivots. If a distance is zero, the search ends, and the corresponding pivot file that has a distance of zero with the vector is retrieved. Otherwise, the search continues by moving to the next node with the lowest distance. For example, if the distance between the vector and pivot 1 is less than the other, the search will proceed in the left subtree of the tree. The search continues until it finds the file with a distance of zero.

7.9 Conclusion

In this chapter, we present the proposed semantic web of Things framework. We detail its layers such as the acquisition layer, semantic layer, fuzzy layer, clustering layer, indexation layer, and application and service layer. The layers also consist of a set of modules that works with each other. We also explain the IoT architecture of the proposed framework and its efficiency for enabling real real-time processing, ensuring scalability, and reducing costs. The framework implements in three cases of the healthcare domain as contributions. The first contribution focuses on the proposed semantic web of things framework to ensure the sharing and alignment across domains through extending SAREF ontology. The contribution addressed integrated health care in public transportation and benefited from IoT paradigms to ensure real-time processing. The second contribution focuses on improving the semantic web of things framework processing by defining a fuzzy ontology from SAREF classical ontology by following the Ikarus methodology. The proposed aims to improve the accuracy of decisions about COVID-19 patients. The framework enables tracking and real-time decision-making by benefiting from IoT architecture inspired by healthcare organizations in Algeria. The third contribution focuses on clustering and indexation layers to store and retrieve RDF data from Big RDF data in a short time. The layers benefit from IoT paradigms and combine fuzzy logic with MEWS to classify RDF data. In addition, indexing the RDF data by using a binary tree includes two pivots and uses distance Euclidean to calculate similar data. in the next chapter, we will present their implementation and results.

Chapter 8

Implementation and experimentation

8.1 Introduction

In this chapter, we will present the implementation and experimentation of the three contributions. The first contribution focuses on developing a semantic web of things framework in health care and public transportation, the second contribution aims to improve this framework using fuzzy logic for monitoring COVID-19 patients, and the third contribution focuses on resolving the problem of storing and retrieving the framework's output that is represented by RDF data. First, we will present the tools we used to perform this work, its interfaces and functionalities for each contribution. Second, we will present a discussion and an evaluation of the results of the three contributions. Finally, we will present a conclusion.

8.2 Implementation of the semantic web of things framework in health care and public transportation

In this Section, we present our prototype implementation of the proposed approach. The implementation takings into consideration the passenger scenario described in section(7.6.3.1).

The implementation starts from developing the ontology to the creation of a dashboard that facilitates the system's use. To build health monitoring public transportation ontology, we have used a free and open-source editor that supports the latest OWL 2 Web Ontology Language and RDF specifications called Protege-5.1.0. It allows us to create our ontology and visualize its components. After creating classes with their properties and relations, we added SWRLTab to apply SWRL rules. We also added a reasoning module represented by the plug-in hermit and pellet, which aims to derive knowledge by exploiting the available data. It also allows us to verify the consistency and correctness of our ontology. The proposed system is implemented in Java programming language. It is supported by its JENA framework [212], an open-source Java framework building semantic web and providing a robust, transactional persistent storage layer and a SPARQL server. To facilitate the use of the proposed system, we have built a system that simulates the smart bus stop. It is designed as a dashboard to show and explain the different steps taken to process the passenger data. The core of this dashboard is based on the Jena, which eases the annotation process according to the proposed ontology for transferring the raw data collected from devices to semantic data in the form RDF file. It allows using this RDF file and other files that consist of personal information, medical profile, previous trips, and historic diagnosis to enrich the knowledge base. It also enables the system to use this knowledge base, the proposed ontology, the customized rules, and the inference engine to deduce knowledge or make a decision. Once the data processing operation is done, the system displays the results represented as a set of services adapted to each passenger's situation. The dashboard in Figure 8.1 uses the SPARQL query to retrieve data from the RDF file for processing and display on its panels. The dashboard has several access points represented by buttons that allow gathering the transmitted data. It has several panels, each of which has a particular function. The "Contact" panel lists all

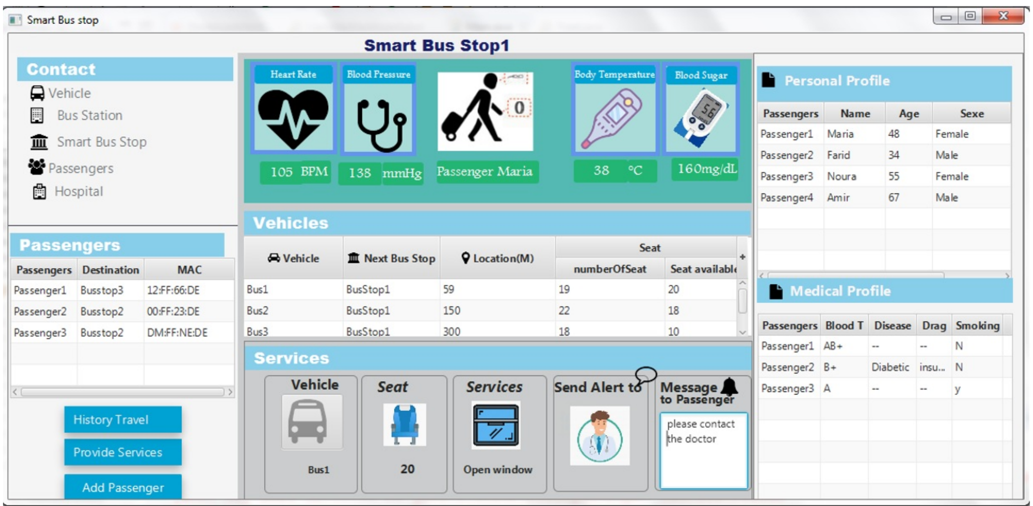

Figure 8.1: Dashboard of our system.

contacts the smart bus stop has been able to store to gather the necessary information from travellers and various vehicles. The panel displays messages that inform passengers, drivers, doctors, etc., as well as information sharing with other smart bus stops, bus stations, and transportation agencies. The "Passengers" panel displays information on passengers, including their names, MAC addresses, and destinations. Additionally, the "Vehicles" panel in the dashboard displays information that lists vehicles near the smart bus stop, the number of people they are carrying, and the number of available seats for more passengers. The two panels, "Medical Profile" and "Personal Profiles," respectively contain information on the passengers' health state and general information. This information helps the system to make accurate decisions. By utilizing the measures shown in the system health metrics panel, we implement the scenario of the passenger Maria from(see section 7.6.3.1) to convert it into reality. Since the passenger Maria has no prior travel history, the system records potential outcomes in the passenger's history file. The system uses the information shown in the previous panels to make decisions based on a variety of services and recommendations. According to the "services" panel, Maria will benefit from the "bus" type vehicle (Bus 1), seat number 20, and an open window, depending on her health condition. A message was sent to Maria asking her to take her medication or call her doctor, as well as a message to her doctors informing them about Maria's health status. We used other cases to validate the system's efficiency with

Figure 8.2: Services provided to passengers has a normal health status

a passenger's data presenting normal health status and another risky health status. The system also provided responses depicted in Figure 8.2, figure 8.3 as a service appropriate to their health status. The system can also be used by doctors to track the health status

Figure 8.3: Services provided to passengers has a risky health status

of their passengers and detect any changes in the health status of their patients.

8.3 Implementation of the semantic web of things using fuzzy ontology for protecting people from COVID-19

The second contribution of our solution improves the accuracy of the proposed framework by using the fuzzy ontology for COVID-19 patient monitoring. To implement it, we used the fuzzy ontology that is developed in the previous chapter (see section 7.7.4.1.2). The fuzzy ontology benefit from the integration of plugin owl2 with protégé 4.3 to enable describing the fuzzy concepts. The fuzzy ontology used for developing a system provides the covid-19 patient with adequate healthy services. To develop a system we use the Java programing language and its tools. We used the framework JENA to annotate and transferred each raw data using the ontology to RDF format. JENA is an open-source Java library developed by the Hewlett-Packard laboratory that allows us the creation and manipulation of OWL ontologies and RDF graphs [212]. It also provides an inference engine allowing reasoning on ontologies. This library eases us for data retrieval and storage of input data using the SPARQL language. To fuzzifier the raw data and enable apply rule base en engine inference, we have used the JFuzzyLogic library [213] for implementing the Fuzzy Control Language (FCL) specifications, which allowed us to easily design fuzzy controllers in the form of FCL files. This library allows the estimation of the degree of membership of the input data based on domain experts. The two libraries integrate into Eclipse editor, in which we realize the different activities like entering the range and type of membership function then transferred the result of the calculation to use with ontology and other files to make the decision. Using their standard fuzzy logic-compliant fuzzy inference method enables us to determine the patient's health state. We developed the graphical user interface shown in Figure 8.4 to simplify the use of the developed framework. The user interface is a dashboard that shows information gathered and loaded from the devices with suggested services. It has a number of screens that display the patient's vital signs, medical conditions, personal information, and the outcome of processing, which includes services regarding the patient's health status. The dashboard also includes a contact files panel that shows the last contact of patients, especially with people who have continuous contact, by displaying information about his/her relationship, location of the contact, etc. The health status and services showed a case when Maria's health status is Abnormal and his colleague in the office has positive COVID. The system suggested contacting the doctor, make test covid-19, taking vitamins and eating healthy food. For enabling the storage of the output of the proposed system, the next implementation eases the storage and accelerates the search time.

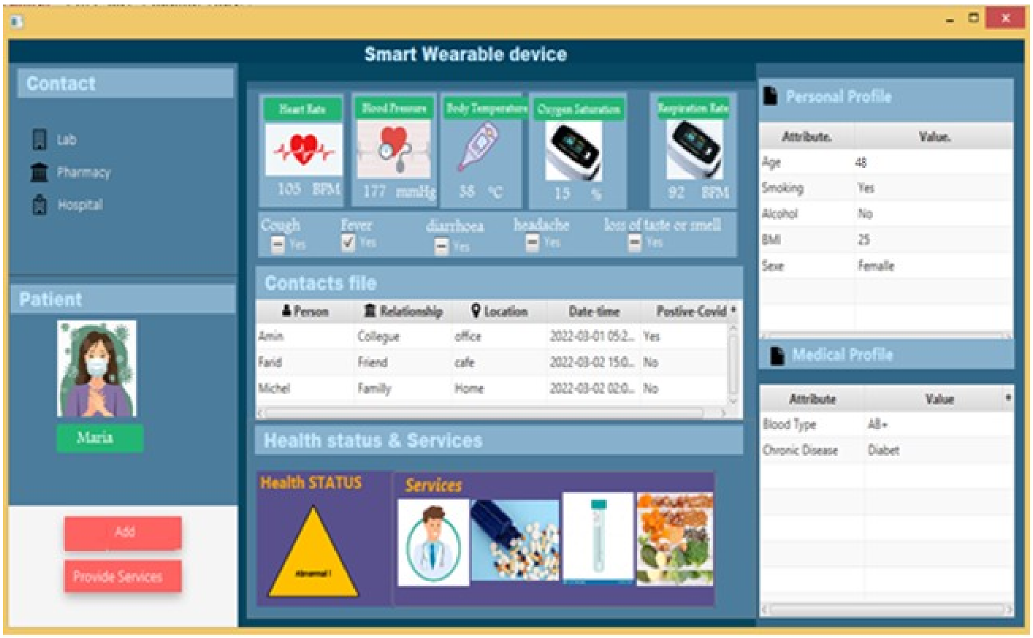

Figure 8.4: Smart wearable device Dashboard for covid-19.

8.4 Implementation of the storage module of Big RDF data generated by the Semantic Web framework

To implement the proposed approach for storing and retrieving the output of the semantic web of things, we use the Python programming language. This open-source software offers a clear and simple syntax that is easy to learn. It is based on a comprehensive standard library that provides a solid base of functionality allowing the programmer to operate more quickly and efficiently. The software enables assessing the feasibility of our approach by implementing a framework on a device running Windows 8.1, which is equipped with an IntelRCore TM i7-6700U CPU, 3.40 GHz, 8 GB Memory, and 500GB ROM. The software provides a skfuzzy library which uses a fuzzy logic algorithm to improve the calculation of patients' health status. As shown in Figure 8.5, the library uses MEWS for the definitions of the membership function of vital signs. The library uses more than three hundred rules to provide 15 classes of health status as result representing the classification of RDF files. The software offers the owlready2 library to generate RDF data for each input from the raw data. The library manipulates the SAREF ontology to represent IoT data in RDF format. Binary trees with two pivots are constructed for the classes that are defined by skfuzzy library for easing storing and retrieving RDF files. We used the Euclidian distance to calculate the distance between the input vector and pivots to determine the similar node. To evaluate the proposed approach, we compare its searching time with a set of

RDF system management also installed. RDFLib is used by implementing its library in Python. Apache Jena [159] and Blazegraph [161] are installed and configured using their plugins. DBGraph [160] application also downloads from its site and is installed on the same computer as the other systems. To verify the proper functioning of this contribution,

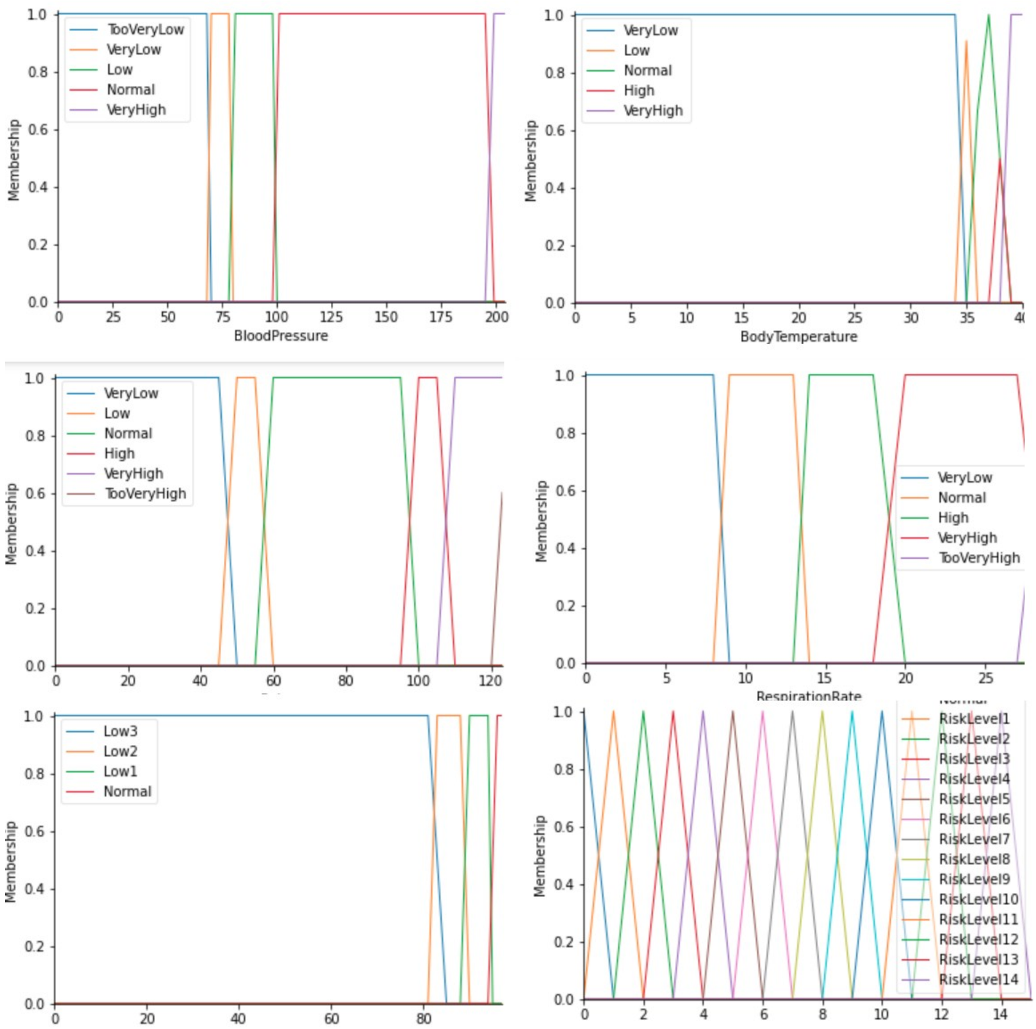

Figure 8.5: Memberships functions

we performed tests. The experimentation and results of these tests are presented in the following section.

8.5 Experimentation: Test the effectiveness of the proposed semantic web of things framework in health care and public transportation

The aim of the first contribution is to provide a solution to the problems of heterogeneity and lack of interoperability in the IoT domain. Implementing the proposed framework in the healthcare and public transportation sectors allows addressing these issues. The proposed framework is based on semantic technologies such as ontology.

8.5.1 Hypothesis

To evaluate the effectiveness of this approach, we propose to test the following hypothesis: "Does the semantic web solve the problems of heterogeneity and interoperability between different IoT applications?". To verify this hypothesis, we should set up an experimental method that makes it possible to compare the performance of our system which is based on ontology and other semantic technologies with that of a decision physician.

8.5.2 Experiment Design

To evaluate the effectiveness of the proposed approach contribution 1, we conducted a test using biometric data from a group of patients. In parallel, these data were shared with doctors to read, process, and give recommendations. Finally, we compared the results of the system's recommendations with those of the physician based on each participant's health situation. The absence of a rich database with several patients with different vital signals and information on the medical and personal profiles does not allow testing the system with all possible cases that present its effectiveness. To overcome this problem, we go through a phase of data collection.

8.5.3 Collection data

The collection of data began by downloading the vital signs data of volunteers provided by the Figshare repository [214]. Secondly, this data was enriched with the help of a doctor by proposing new metrics representing critical situations that passengers may encounter, in addition to the medical profile information such as gender, age, etc., used to test the system. Finally, the data was verified by a physician who presented his acceptance that this biometric data could cover as many cases as possible. The data set presented in Table 8.1 shows 30 patients with measurement values of five vital signs: blood pressure, body temperature, SPO2 (oxygen saturation), and heart rate. The table also presents values related to personnel profiles such as age, Gender, Weight, Height, and medical profile values concerning the drinking of alcohol, drug, and smoking.

8.5.3.1 Experimental Methodology

To evaluate the effectiveness of our proposal, we conducted an experimental test with a doctor. The objective of this experiment is to verify the impact of the system results with the doctor's recommendations. Dr A. KIRATI, an expert in cardiovascular diseases from the Algerian city of Guelma, supervised the performed tests. Firstly, the doctor received the dataset to check if it covered as many scenarios as possible. She was satisfied with its richness, noting that it was available for testing the system. Secondly, The doctor also gets a questionnaire that asks to determine the type of car to suggest, such as a bus, car, or emergency car. It also asks for supplementary services such as refreshments and air conditioning, as well as asking the doctor to specify the direction, should be informed by SMS, such as hospital, passengers, or doctors. After the doctor examined and analyzed the data attached to the health of the passengers, she fulfilled the questionnaire. Finally, we compared the doctor's recommendations with those of the system.

8.5.3.2 Experimental Results and Discussion

The recommended services can classify into four types of transportation services. The passengers with normal health status benefit from type 1 services, which cover a bus and a chair. In addition, passengers with abnormal health status that do not cause worry to benefit from type 2 services, including a bus, a chair, and some additional services such as window opening and medication reminders. However, the rest of these passengers benefit from the type 3 benefits of a car and a letter to the doctor. The passengers with a risky health status benefited from type 4 services, which are emergency cars and messages to the hospital. From the results of the experiment, shown in Figure 8.6, the physician advised 30% of the 30 passengers to use a bus (service 1), 3.3% to use a bus with additional services (service 2), and 16.6% to use a car to get around (service 3). The physician recommended that 50% of patients use an ambulance for emergency assistance (service 4). For the same dataset of 30 passengers, the results of the developed system were as follows: the system advised 30% of the passengers to take a bus (service 1), 6% to take a bus with additional services (service 2), 24% to take a car with contact with doctors (service 3), and 33% to take an ambulance in case of emergency (service 4).

As shown in Figure 8.7, three categories occur when comparing the two results of the experiment. The first category indicates that there is conformity between the two results

Passengers	Heart Rate	Blood Pressure	Body Temperature	Spo2	Respiratory Rate	Age	Gender	Weight	Height	Alcohol	Smoking	Drug
P1	98	109/73	37	96	15	22	1	61	64	1	1	1
P2	193	140/90	36,7	98	11	38	1	119	171	1	1	2
P3	91	106/68	36,5	97	18	26	1	113	177	1	1	1
P4	98	109/73	36,3	99	18	19	2	58	161	1	1	1
P5	99	90/60	37,2	97	12	20	2	60	168	1	1	1
P6	80	112/72	36	97	29	20	2	65	151	1	1	2
P7	103	122/78	37,2	99	20	28	1	72	164	1	2	1
P8	73	104/67	36,8	98	26	75	2	51	157	1	2	1
P9	72	180/97	39,3	94.3	20	80	1	98	166	1	1	1
P10	116	190/117	38,7	95.2	22	85	2	86	160	2	1	2
P11	81	107/64	37,1	77	14	37	2	64	160	1	1	1
P12	90	140/90	40,7	98	24	43	1	119	173	1	1	1
P13	97	114/62	36,8	97	16	20	2	55	145	1	2	1
P14	92	99/63	39,3	89	10	26	2	74	164	1	2	1
P15	80	126/78	37,2	97	20	67	1	65	166	2	2	1
P16	103	113/73	36,4	98	20	20	1	94	170	2	1	2
P17	100	152/90	38,2	99	22	55	2	61	167	1	1	1
P18	81	104/68	34,8	98	16	26	1	74	176	1	1	1
P19	98	159/97	35,5	96	25	23	1	78	182	1	1	1
P20	193	117/62	38,4	96	23	24	1	103	184	2	1	1
P21	98	100/61	40,1	94.4	28	66	2	62	148	2	2	1
P22	81	81/55	36,9	99	15	28	1	55	142	2	1	1
P23	91	110/61	38,9	94.5	24	30	2	75	136	2	1	1
P24	98	125/66	39,3	95.4	20	35	1	60	130	1	2	1
P25	109	132/61	40,7	94.4	21	40	2	65	124	1	1	1
P26	81	139/86	38,1	95.3	12	45	1	80	118	1	1	2
P27	103	137/90	34.5	94.5	23	50	1	74	112	1	1	1
P28	77	95/86	37.1	97.1	10	55	2	58	106	1	1	1
P29	72	119/77	36	96	19	60	1	72	100	1	1	1
P30	113	160/99	39.5	97.2	11	65	1	63	119	1	2	2

Gender = 1 signifies a male patient, whereas Gender = 2 shows a female patient. Alcohol = 1 denotes alcohol consumption, whereas 2 indicates does not drink. Smoking = 1 indicates a patient who smokes, whereas 2 indicates a patient who does not smoke. Medication =1 indicates the presence of medication, whereas 2 indicates the avoidance of medication.

Table 8.1: Dataset of vital signs of patients

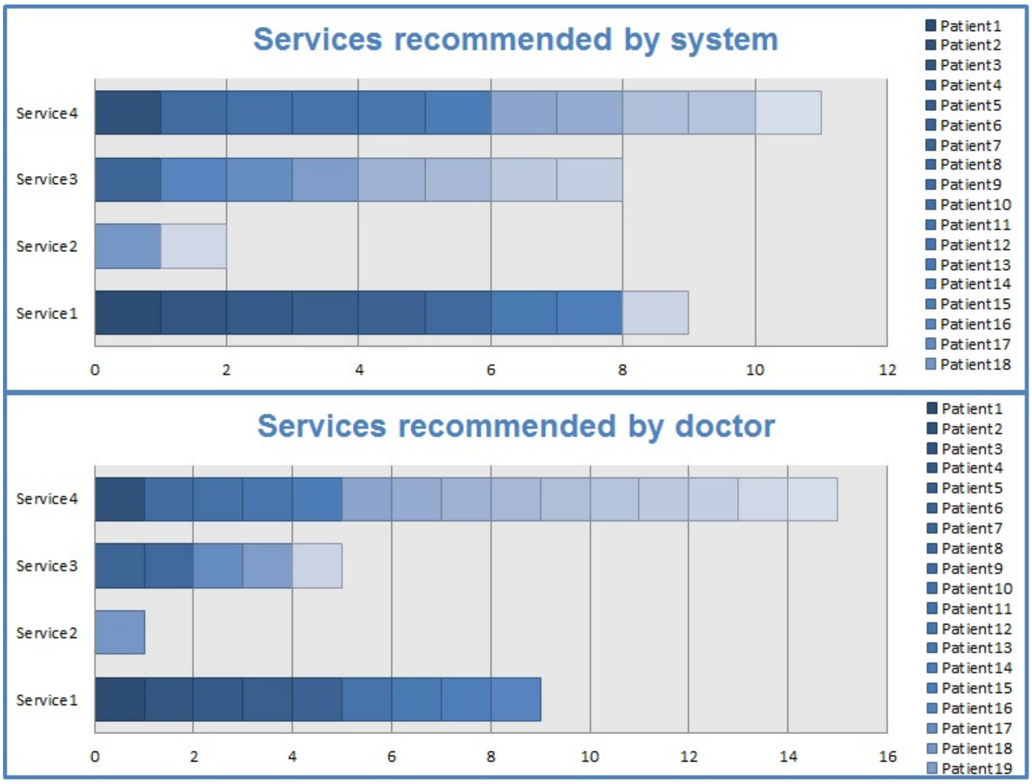

Figure 8.6: Comparison between the proposed system and the physician recommendation services.

at 70%, the second category shows the convergence of the services offered without danger to the health of the passengers at 27%, and the third category shows a divergence at 3%. According to the experimental results, the conformity rate between the results of the proposed approach and the doctor's recommendation is high (70%), proving the proposed system's efficiency in offering services is confirmed to what the doctor recommends. Results also demonstrate that the system reasoning is closer to the doctor's decision-making despite the heterogeneous sensor data collected in various domains. The result also shows that the system has reached a convergence of 27% of the doctor's recommendations in the treatment of the cases characterized by duplication, which requires optimizing the levels of treatment in the ranges of vital signals to obtain maximum accuracy. The 3% rate of incorrect suggestions may return to several causes (measurement errors, incorrect SWRL rules, etc.), all of which can be verified by applying more experiments or providing a preprocessing of data. Generally, the results show that the proposed system based on semantic data processing succeeded in satisfying the physician's performance. It has succeeded in overcoming the heterogeneity of data and domains, demonstrating its capability to achieve semantic interoperability. The system can exchange the heterogeneous IoT data

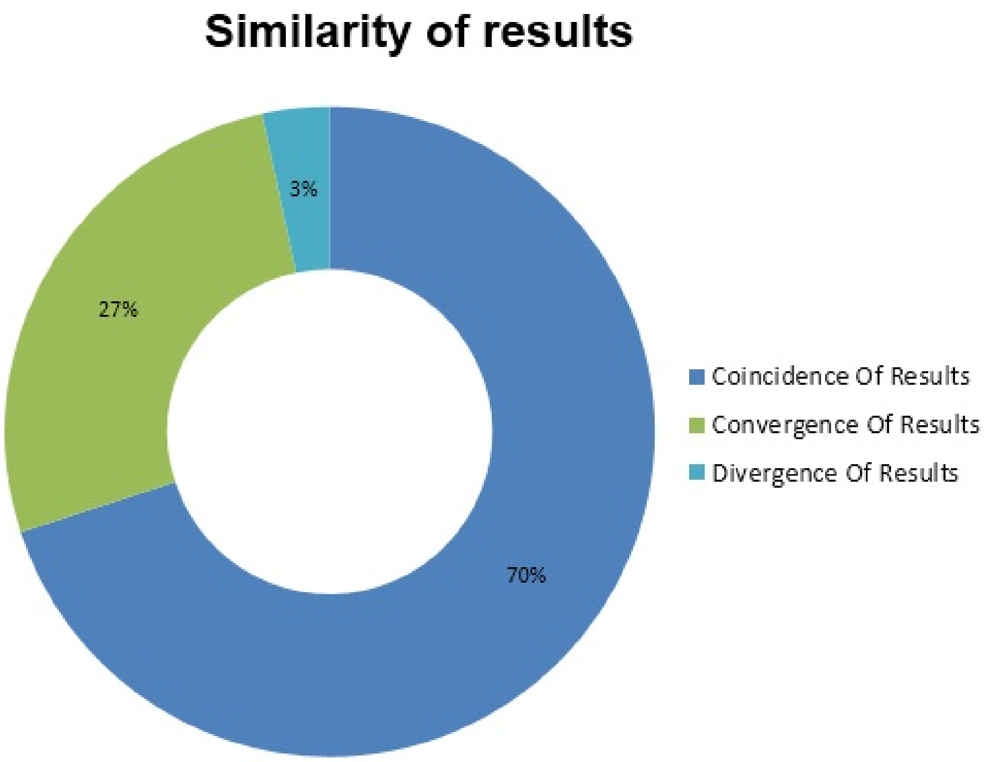

Figure 8.7: Calculation of the similarity of results.

produced by the two domains to build new knowledge that can be used to anticipate the health of passengers. Doctors or hospitals can use this system for disease surveillance during outbreaks like COVID-19 or analytical research on disease spread patterns.

8.6 Experimentation2: testing the effectiveness of the system based on the semantic web of things and fuzzy ontology to provide accurate results

The second contribution integrated fuzzy logic in the semantic web of things to enable more realistic and accurate modeling of knowledge and concepts that are naturally imprecise or ambiguous. It has the objective of improving the accuracy and reliability of decision-making about the health status of COVID-19 patients.

8.6.1 Hypothesis

In order to determine the effectiveness of the Approach, we focus on testing the following hypothesis: "Does the integration of fuzzy ontology in the semantic web framework improve the accuracy of results?" An implementation of the integration of fuzzy logic with the semantic web of things was conducted. The implementation represents a system that enables to load of vital signs data and provides users with a recommendation about their health status. The system results can be compared with those of a traditional system or a doctor's decision.

8.6.2 Experiment Design

In order to evaluate the effectiveness of the proposed approach contribution2, we conducted a test using data collected from a group of COVID-19 patients. The data were also disseminated to physicians for consultation, review, and recommendation. Then, the system's recommendations were compared with the physician's recommendations based on each participant's health status, as well as with the results of the work done in NEWS-2.

8.6.3 Collection Data

The Pondicherry Institute of Medical Sciences provides a dataset of 399 hospitalized suffering from COVID-19 [215]. The published dataset consists of vital signs measurements such as body temperature, pulse, blood pressure, respiration rate, and oxygen saturation. It has personal information like age high, weight, alcohol, and smoking. It also includes the medical profile like chronic disease, covid-19 symptoms, and other data.

8.6.4 Experimentation methodology

To evaluate the proposed system, we followed a methodology based on the assistance of a team of Algerian physicians to supervise this experiment using previously collected data. Firstly, Before starting this experiment, the four physicians reviewed these data and confirmed that this dataset covers a maximum of instances where they found it very rich. Secondly, Physicians were given a data set of 200 patients and a questionnaire that asked them to make a diagnosis of the patient's health based on this dataset. The physicians were asked to present the patient's health status between normal, abnormal, and risky and to recommend treatments. After examining the data, the physicians responded to this questionnaire. Finally, the doctors' answers were then compared with the results of the proposed system and the Pugazhvannan results [215].

8.6.4.1 Experimentation results

Figure 8.8 shows the questionnaire's results, which show that 27% of all patients have normal health statuses and that 38.5% have abnormal health statuses and had recommendations to perform a COVID-19 test. In addition, the doctor advised them to take their medication or/and to avoid drinking alcohol and smoking. The last case is the patient at risk, they represent 34.5% of patients who would need sophisticated care, such as urgent evacuation by ambulance hospital or directly going to the hospital. The Pugazh-

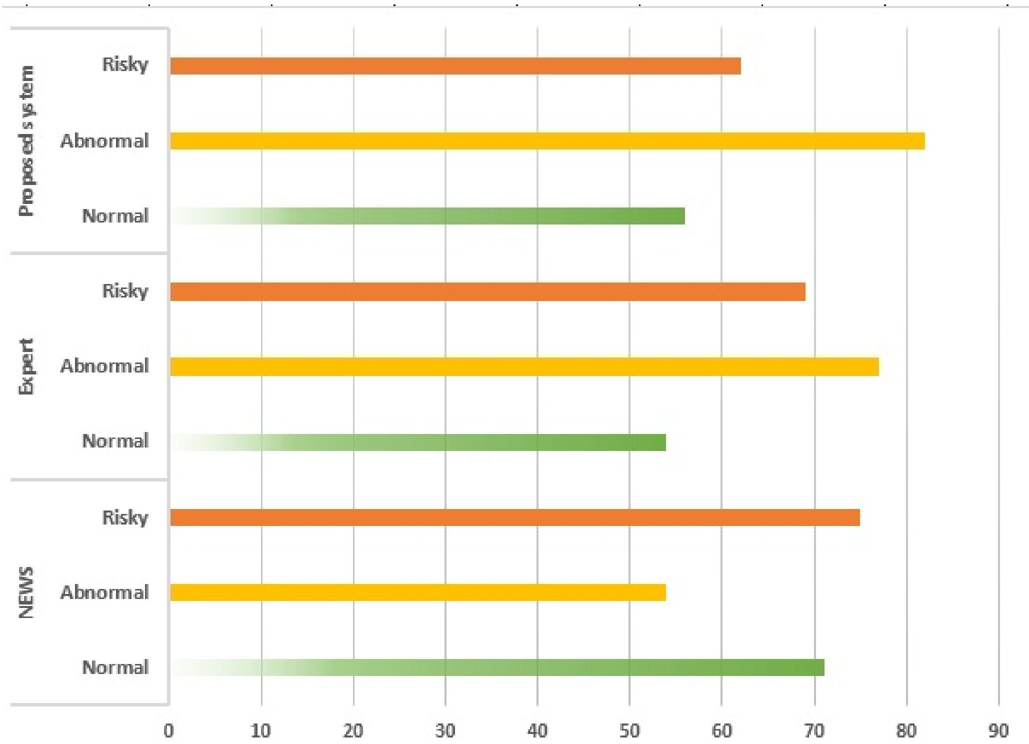

Figure 8.8: Results of proposed approach, doctors and Pugazhvannan's work.

vannan uses the NEWS to generate a score for each patient using the same dataset. After we have translated these results into linguistic variables we found: that 35.5% of patients have normal health status, 27% have abnormal health, and 37.5% have risky health status. Using the same patient dataset, the proposed approach found that 28% of patients have normal health status, 42% have abnormal health status, and 30% are at risk. The comparison between the system result, Pugazhvannan work and the experimental results can be divided into three categories as illustrated in Figure 8.9. First, we compared system results with the experimental results of doctors. There is an 87% conformity between the two outcomes and there is an 8% convergence of patients' health status and a 5% divergence. The results of Pugazhvannan's are compared to those of doctors and finding

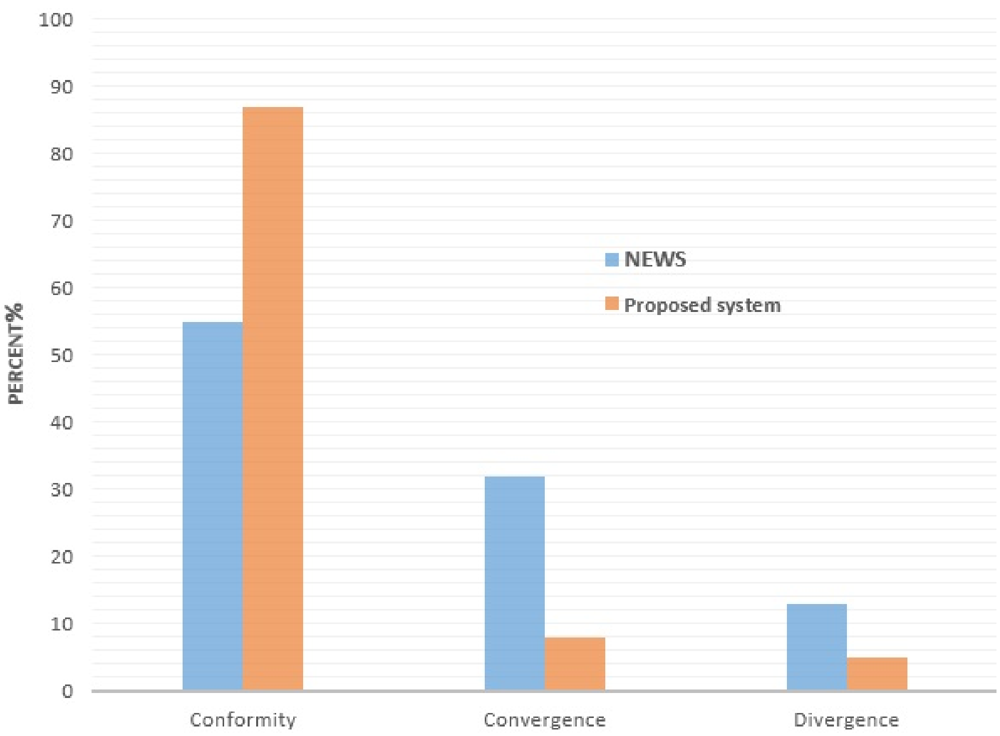

Figure 8.9: Calculation of the similarity of results.

the coincidence, convergence, and divergence are 55, 32, and 13 respectively.

8.6.5 Discussion

Based on the result of the experiment, we can infer some important conclusions:

- The proposed approach satisfied a great coincidence with the doctor's diagnosis, This is what qualifies him for following the patients of COVID-19 for protecting them from the virus.
- The proposed system uses the NEWS score system in the calculation of membership function and can improve the results in comparison to the work of Pugazhvannan.
- The system is based on fuzzy ontology, which allows the integration of information about chronic diseases and symptoms, in addition to vital indicators, which allows the use of different resources to get an accurate result.
- As this dataset is large and includes a variety of potential instances, we can conclude that the system is capable of producing positive outcomes for patients in different phases of infection.

- The proposed system uses patient contact and simplifies the processing of vital sign changes that ease the rapid identification of COVID-19.
- The use of fuzzy logic through fuzzy ontology improves the semantic web of things framework.

8.7 Experimentation3: Testing the effectiveness of the proposed approach for storing and retrieving RDF data in a short time

The clustering layer enables the grouping of the RDF files in groups that reduce the space searching. The indexing layer also avoids the search in all data. The proposed framework that is based on the two layers aims to store its output in RDF format and retrieve it very quickly.

8.7.1 Hypothesis

To evaluate the effectiveness of our approach, we focus on the validation of the following hypothesis: How does storing framework results in RDF format through the indexing and grouping layers make it easier to store and speed up information retrieval?

8.7.2 Experiment Design

To verify this hypothesis and provide an evaluation of the framework proposed in Contribution 3, we followed a series of steps. First, we started by generating RDF files from the raw data collected, then we classified them into classes. Then, construct the tree by inserting the files. Finally, we asked some questions to estimate and compare the search time of the proposed approaches with other systems.

8.7.3 Data collection

Since there is a lack of IoT sensor data related to the health status of patients, we had to use a dataset that contains the measurement values of patients who were monitored in the hospital. The MIMIC dataset is an open-access collection of anonymized patient information from a hospital in the United States [216]. The MIMIC dataset consists of a set of measurements of patients' vital signs. Each measurement is represented by blood pressure, oxygen saturation, heart rate, and oxygen saturation values. However, the dataset is in waveform format. BIDMC is a dataset taken from MIMIC to express its data in CSV format [217]. Moreover, the BIDMC includes the patient's information, such as age, gender, and other details.

8.7.3.1 RDF files generation

Base on SAREF ontology the approach enables to generate RDF files as depicted in Figure 8.10 three datasets of RDF files can be created from the obtained data using the approach. These datasets are used to test the methodology with various data sizes. The first dataset has 6000 files; the second has more than 13000 files and the third has more than 25,000 RDF files.

```xml
<?xml version="1.0"?>
 <rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:HTP="http://www.semanticweb.org/HTP#" >
<rdf:Description rdf:about="http://www.semanticweb.org/HTP#Andrew_Stoltz">
<rdf:type rdf:resource="http://www.semanticweb.org/HTP#Patient"/>
<HTP:hasAge rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">88</HTP:hasAge>
<HTP:Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Andrew Stoltz</HTP:Name>
<HTP:Sexe rdf:datatype="http://www.w3.org/2001/XMLSchema#string">male</HTP:Sexe>
<HTP:hasDevice>
<rdf:Description rdf:about="Device">
<HTP:consistOf>
<rdf:Description rdf:about="http://www.semanticweb.org/HTP#HR_sensor">
<HTP:MakeMeasurement>
<rdf:Description rdf:about="http://www.semanticweb.org/HTP#Measurement0">
<HTP:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">94</HTP:value>
<HTP:time rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">6</HTP:time>
</rdf:Description>
</HTP:MakeMeasurement>
</rdf:Description>
</HTP:consistOf>
<HTP:consistOf>
<rdf:Description rdf:about="http://www.semanticweb.org/HTP#Pulse_sensor">
<HTP:MakeMeasurement>
<rdf:Description rdf:about="http://www.semanticweb.org/HTP#Measurement1">
<HTP:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">94</HTP:value>
<HTP:time rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">6</HTP:time>
</rdf:Description>
</HTP:MakeMeasurement>
</rdf:Description>
</HTP:consistOf>
<HTP:consistOf>
<rdf:Description rdf:about="http://www.semanticweb.org/HTP#RESP_sensor">
<HTP:MakeMeasurement>
<rdf:Description rdf:about="http://www.semanticweb.org/HTP#Measurement2">
```

Figure 8.10: An example of RDF file

8.7.3.2 Evaluation and comparison of Clustering without and with fuzzy logic

Classification allows grouping RDF files into groups. Using the global score provided by MEWS, we classified each file in a class from 15 classes. A fuzzy logic algorithm is also used to improve this classification. Through these results and the figure 8.11, the approach has succeeded in improving a thousand grouping of similar RDF which means it can help the physicians have a better understanding of the patient's status and take a good decision.

Figure 8.11: Classification using MEWS without and with fuzzy logic

8.7.3.3 The Evaluation and comparison of the index construction using binary tree

To evaluate the construction index for the Binary tree with two pivots, we use some criteria such as the number of computed distances and the construction time (as shown in Figure 8.12) criteria. number of computed distances (number of comparisons) represents the calculation of the comparison of the distance of the new vector with the two pivots during the creation of the tree, which is increased according to the growth of the size of the dataset. The construction time also increased with the growth of the dataset as a result of the increase in numbers comparison made. Figure 8.13 depicts the binary tree's

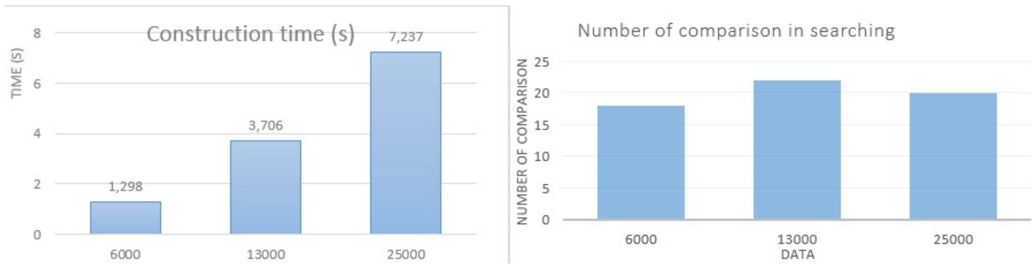

Figure 8.12: Construction time and number of comparison

height and the number of leaves. The tree has leverage height and the number of leaves has increased gradually with the size of the data which indicates the tree is balanced and well-built.

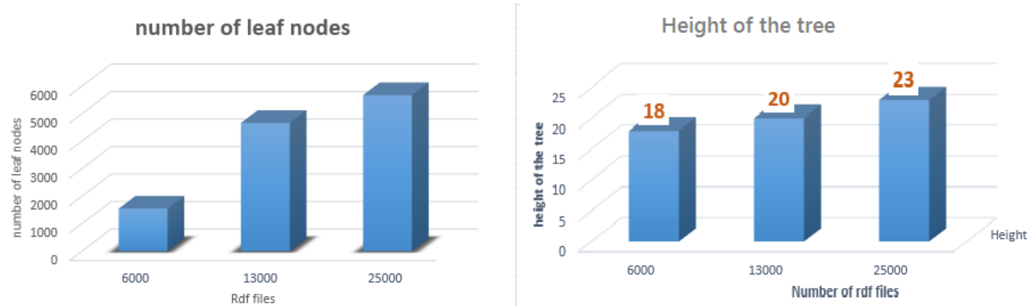

Figure 8.13: Evaluation of the index structure.

8.7.3.4 Searching with and without the tree

RDF files contains IoT data measurement collected from a body of patients and medical and personnel information in addition to the result of diagnosis and specific recommendations to this health status. Retrieving and reusing these files is very important and can help the healthcare domain for providing adequate services at the appropriate time. As searching time is important to retrieve similar RDF files, We need to evaluate the searching time of this approach. To achieve it, we conducted an experiment consisting of 10 queries to retrieve the RDF file and its similarities with and without using binary tree indexing and evaluated its effectiveness. The queries represent measurement values that have been selected randomly from this RDF files dataset. By comparing the results shown in Figure 8.14, we found that searching using a binary has a short search time as a result of a reduction in the number of comparisons. Furthermore, the indexing layer benefits from the clustering layer by utilization of the fuzzy algorithm in constructing the Binary tree by facilitating the search space.

8.7.3.5 Comparison of the search time of the proposed approach with other systems

Jena, Rdflib, Blazegraph, and DBgraph are applications of system management of RDF data using a triple store to store RDF data. To retrieve the data using these systems, the user needs to ask by SPARQL query. Using a set of queries to retrieve RDF data, we asked the system and our approach. To evaluate the searching time comparing the systems with different sizes of data, we use the three dataset. The results obtained from

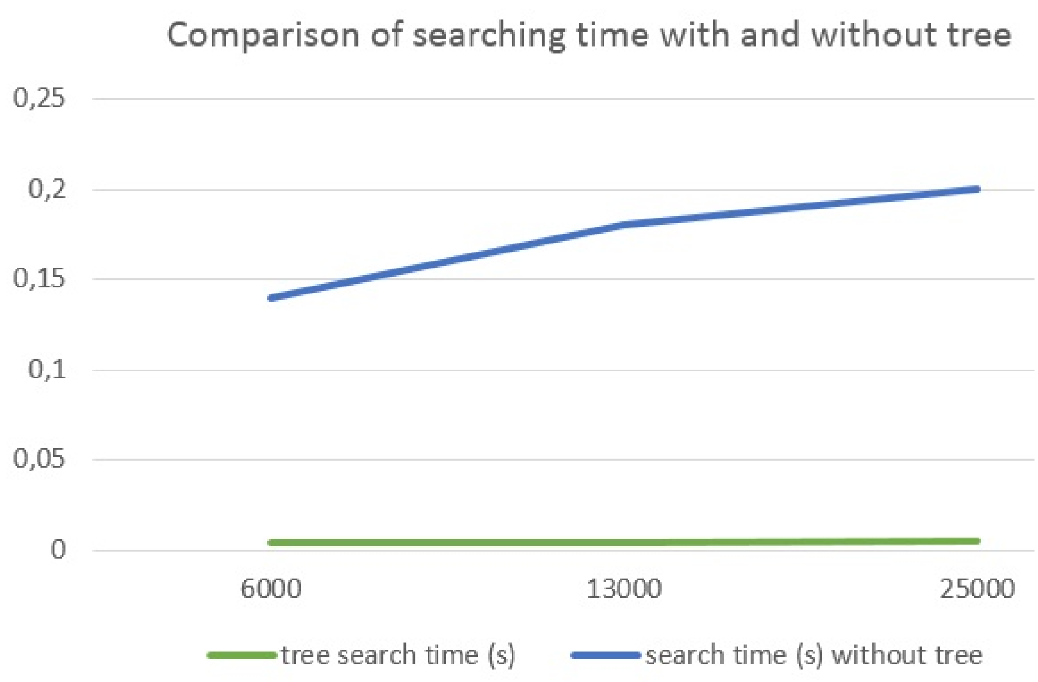

Figure 8.14: Search time with and without the tree.

experience depicted in Figure 8.15 show the proposed approach has a short time to search and retrieve RDF data. Different to Jena, Rdflib, and Blazegraph, the search time of the proposed approach was not significantly affected by the data size.

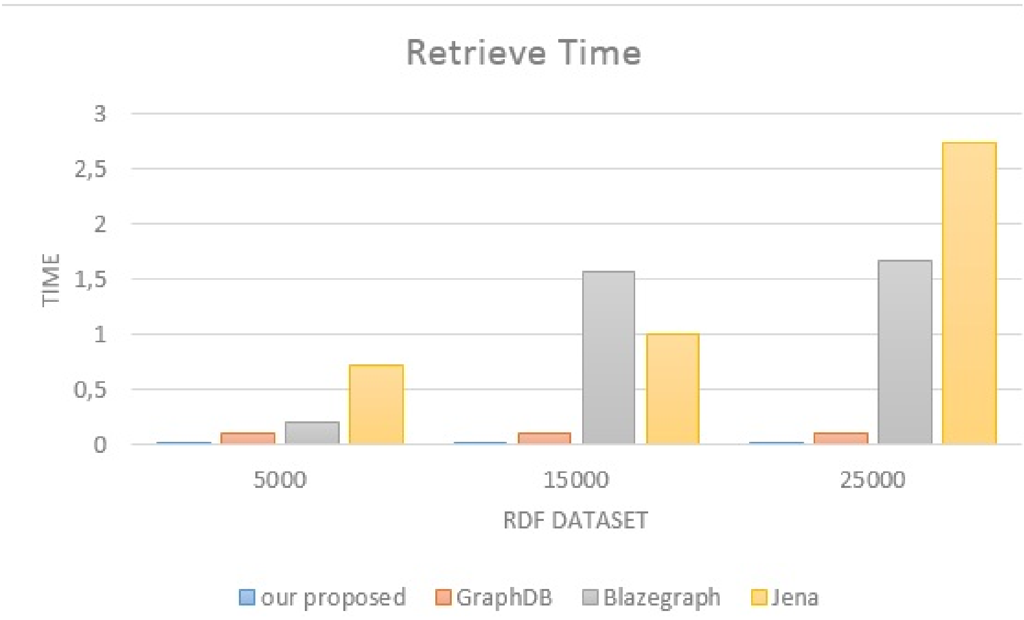

Figure 8.15: Comparison of the proposed framework with the RDF systems management.

after all these evaluations, The proposed system proves its capabilities to generate RDF files, classify into accurate classes and then store them with structure ease to retrieve information very quickly. The proposed system facilitates use by avoiding any experience in SPARQL language to retrieve the information. The proposed approach can ensure tracking of the health of patients, provide them with accurate recommendation, and helps the doctors and hospital.

8.8 Conclusion

In this chapter, we examined three contributions to the field of the semantic web of things. We present the implementation of the three contributions. We also present experimentation of the first contribution to proving its ability to provide semantic interoperability across domains by making decisions that are confirmed to doctor recommendations. The results show the proposed approach achieves results close to the doctor's decision and can be used in different domains and other diseases. After experimentation of the second contribution, we concluded the proposed approach to improve the semantic web of things framework using fuzzy ontology proves its efficiency in providing a correct diagnosis by determining patients' health status and recommended services. We concluded also the proposed approach improves the precision and accuracy that making the framework more efficient. For the third contribution, the experimentation showed the efficiency of the proposed approach to generate, store, and retrieve RDF data with IoT health data. The proposal is based on fuzzy logic that improves the clustering using MEWS. It is also based on a well-constructed and balanced binary tree with two pivots. The approach achieves a short time to retrieve information compared to other systems like Jena and Blazegraph.

Conclusion and Future Work

The Internet of Things(IoT) is defined as a technology consisting of things equipped with sensors, software, and connection that allows them to collect and exchange data with other things and systems over the Internet. The Internet of Things is destined to help humans in their daily activities. Its ability to observe the physical world and provide information for decision-making has the potential to revolutionize many areas, such as healthcare. It can increase the operational efficiency of these domains by performing repetitive activities to reduce the number of workers, free up human interest for more relevant activities, reduce costs, and provide other benefits. However, implementing IoT technology faces many significant challenges, including heterogeneity and lack of interoperability. Heterogeneity refers to the differences among IoT systems and devices, such as variances in their architectures, protocols, and technical specifications, which can lead to challenges in integrating them and result in a lack of interoperability. This heterogeneity can result in difficulties in integrating devices and systems due to the lack of interoperability between IoT devices and applications. Interoperability enables the different IoT systems and technologies to work together and exchange data seamlessly, which improves IoT domain efficacity. Semantic interoperability is a specific type of interoperability that refers to the ability of diverse systems and devices to share data in a meaningful and understandable way with all parties involved. Integrating the semantic web allows semantic interoperability and led to the Semantic Web of Things (SWoT). The latter aims to provide a common framework for representing and exchanging information between IoT devices and systems. The semantic web of things is based on the use of ontologies, which are formal representations of knowledge and concepts that can play a critical role in its efficiency. However, current SWoT frameworks suffer from a set of limitations, most of them are associated with ontology:

- The ontologies have difficulties in sharing and aligning data with others, especially data from many domains.

- The ontologies cannot represent and share the vagueness and imprecise data that represent the healthcare domain.

- The frameworks are missing a mechanism to enable the storage and retrieval of RDF IoT data very quickly.
- The frameworks have difficulties adapting to the increasing number of connected devices increases and ensuring real-time processing.

This thesis centered around the creation of the semantic web of things framework to address the previous challenges and facilitate data processing in the IoT domain. The ultimate goal was to create a more efficient and effective IoT system by leveraging the capabilities of the semantic web of things. The framework is implemented in the healthcare domain to improve the quality of life of a person and provides health services. The first part of the thesis elucidates the subject of health care and internet of things technology and presents in-depth explanations of the semantic web, fuzzy logic, and big data technology. In addition, the thesis provided a comprehensive analysis of the most recent developments in the semantic web of things. In the second section of this thesis, three contributions that aim to improve the semantic web framework's ability to deal with heterogeneous data and achieve semantic interoperability were described. These contributions also aimed to increase the effectiveness of data processing, resulting in faster processing times and more precise outcomes.

First, the thesis described how the semantic web of things could handle the heterogeneity, achieve semantic interoperability, and ensure alignment and sharing, especially between different ontologies from different domains. The ontology was developed in three steps following the neon methodology that enables the creation of a new ontology, extending the SAREF ontology for wearable devices, and merging the two ontologies with some modifications. The framework is implemented to manage the integration of public transportation and health monitoring. It also exploits semantic web technologies such as SWRL rules, inference and query engines to process data for decision-making and deliver health services. It is based on an IoT architecture that enables data to be collected and processed in real time. The architecture is based on a smart bus stop functioning as an Edge, interconnected bus stations as a Fog, and the transport agency as a Cloud.

Second, in order to improve the processing of the semantic web of things framework and ensure accurate decisions on IoT data, the integration of fuzzy logic into the framework was proposed. The need for this integration arises from IoT data's imprecise and vague nature, which can lead to incorrect processing results. This integration involves defining a fuzzy ontology that combines the fuzzy theory concept with the classical ontology. The framework proposed is implemented in health monitoring to safeguard individuals against COVID-19 infection. Following the Ikarus methodology, the fuzzy ontology was created from classical ontology represented by SAREF4wear and enriched with concepts and relationships related to COVID-19 and the health domain. The proposed framework is based on an IoT architecture inspired by health organizations in Algeria to enable real-time diagnostics and ensure the right IoT function with the required resources.

Third, To enable the storage and retrieval of results of the semantic web of things framework, we have proposed a framework model that integrates the generation, storage and retrieval of RDF data. The framework utilizes an IoT architecture consisting of IoT devices, Edge, Fog, and Cloud layers. The fog layer is divided into two layers, where the output of one layer serves as the input of the other. The fog level 1 enables the clustering of RDF files that generate from the edge layer into groups that are improved by a fuzzy logic algorithm. The fog level2 allows the indexation of the RDF files using a binary tree with two pivots by calculating the Euclidean distance. The Framework is implemented to monitor patient health using the MEWS score system and to group patients according to their health status.

To validate the theoretical approaches proposed in this research context, we implemented these three contributions to verify their capabilities to provide patients with appropriate health services in adequate time. We performed three experiments using data representing patients' vital signs and with the help of doctors.

In the first experiment, under the supervision of Dr. A. KIRATI, a specialist in cardiovascular diseases in Guelma, Algeria, we compared the results of the proposed approach with those of the physician to evaluate its effectiveness. The experiment's results show that the proposed system and the doctor's diagnostic results convergent. Unfortunately, because this data collection is incomplete, we could not compare the proposed approach with too many patients.

In the second experiment, which involved larger data and was supervised by four doctors, the performance of the proposed approach was compared to physician recommendations and another work of Pugazhvanna that were based on the National Early Warning Score2 (NEWS2). Compared to the method that used simply NEWS2, the experiment's results showed that the system's outcomes aligned more with the doctors' recommendations.

The third experiment involved utilizing vital signs measurement data to create RDF files as framework output. The RDF files was clustered using MEWS and Fuzzy logic and indexed using a binary tree. We evaluated the clustering by comparing the classification between the MEWS system with and without fuzzy logic. We calculate the binary time construction, a number of comparisons, and other calculation methods to evaluate the indexation. In addition, the retrieve time of the approach is compared to the retrieve time of some RDF management systems. The experiment's results show that fuzzy logic improves the classification of RDF files, and indexation reduces retrieval time by reducing search spacing.

In future work, we intend to enhance the framework by adding a pre-processing phase to improve the data quality, avoid the faulty diagnostic state, detect missing data, and eliminate spurious data. We also intend to use other indexing and clustering algorithms to enable more efficient organization and faster retrieval of relevant information. In addition, we intend to use the framework in other domains, such as protecting agriculture plants from diseases, up-to-date weather forecasting, etc. As the proposed framework addresses the heterogeneity and lacks interoperability challenges, We plan to improve the framework for handling other IoT challenges like security. We will use the semantic web to enhance security in IoT environments by facilitating authentication, authorization, identity management, attribute-based access control, anomaly detection, and secure data integration.

Bibliography

[1] Bastien Confais. *Conception d'un système de partage de données adapté à un environnement de Fog Computing.* PhD thesis, Université de Nantes, 2018.

[2] Carlos E Palau, Giancarlo Fortino, Miguel Montesinos, George Exarchakos, Pablo Giménez, Garik Markarian, Valérie Castay, Flavio Fuart, Wiesław Pawłowski, Marina Mortara, et al. *Interoperability of Heterogeneous IoT Platforms.* Springer, 2021.

[3] Dominique D Guinard and Vlad M Trifa. Building the web of things. vol. 3, 2016.

[4] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.

[5] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[6] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2011.

[7] saref ontology. https://saref.etsi.org/core/v3.1.1/, 2020. last accede 2023.04.25.

[8] Christos Anagnostopoulos and Stathes Hadjiefthymiades. Advanced fuzzy inference engines in situation aware computing. *Fuzzy sets and systems*, 161(4):498–521, 2010.

[9] Hanêne Ghorbel, Afef Bahri, and Rafik Bouaziz. Fuzzy protégé for fuzzy ontology models. *Age*, 12(18):30, 2009.

[10] Panos Alexopoulos, Manolis Wallace, Konstantinos Kafentzis, and Dimitris Askounis. Ikarus-onto: a methodology to develop fuzzy ontologies from crisp ones. *Knowledge and information systems*, 32:667–695, 2012.

[11] Syed Ahmad Chan Bukhari, Ali Kashif Bashir, and Khalid Mahmood Malik. Semantic web in the age of big data: a perspective. 2018.

[12] Zineddine Kouahla, Ala-Eddine Benrazek, Mohamed Amine Ferrag, Brahim Farou, Hamid Seridi, Muhammet Kurulay, Adeel Anjum, and Alia Asheralieva. A survey on big iot data indexing: Potential solutions, recent advancements, and open issues. *Future Internet*, 14(1):19, 2021.

[13] Abdelhalim Hadjadj and Khaled Halimi. An integration of health monitoring system in public transport using the semantic web of things. *JUCS: Journal of Universal Computer Science*, 27(12), 2021.

[14] Abdelhalim Hadjadj and Khaled Halimi. Covid-19 patients' health monitoring system using fuzzy ontology and internet of things. *Iraqi Journal For Computer Science and Mathematics*, 4(1):191–203, 2023.

[15] Louise S van Galen, Casper C Dijkstra, Jeroen Ludikhuize, Mark HH Kramer, and Prabath WB Nanayakkara. A protocolised once a day modified early warning score (mews) measurement is an appropriate screening tool for major adverse events in a general hospital population. *PLoS One*, 11(8):e0160811, 2016.

[16] N Alam, IL Vegting, E Houben, B Van Berkel, L Vaughan, MHH Kramer, and PWB Nanayakkara. Exploring the performance of the national early warning score (news) in a european emergency department. *Resuscitation*, 90:111–115, 2015.

[17] James Manyika, Michael Chui, Peter Bisson, Jonathan Woetzel, Richard Dobbs, Jacques Bughin, and Dan Aharon. Unlocking the potential of the internet of things. *McKinsey Global Institute*, 1, 2015.

[18] Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski, Paweł Szmeja, and Katarzyna Wasielewska. Semantic interoperability in the internet of things: An overview from the inter-iot perspective. *Journal of Network and Computer Applications*, 81:111–124, 2017.

[19] Miriam Rovesti, Massimo Fioranelli, Paola Petrelli, Francesca Satolli, Maria Grazia Roccia, Serena Gianfaldoni, Georgi Tchernev, Uwe Wollina, Jacopo Lotti, Claudio Feliciani, et al. Health and illness in history, science and society. *Open Access Macedonian Journal of Medical Sciences*, 6(1):163, 2018.

[20] Eleanor M Winpenny, Celine Miani, Emma Pitchforth, Sarah King, and Martin Roland. Improving the effectiveness and efficiency of outpatient services: a scoping

review of interventions at the primary–secondary care interface. *Journal of Health Services Research & Policy*, 22(1):53–64, 2017.

[21] Rachel C Shelton, Brittany Rhoades Cooper, and Shannon Wiltsey Stirman. The sustainability of evidence-based interventions and practices in public health and health care. *Annual review of public health*, 39:55–76, 2018.

[22] Parisa Mokhatri-Hesari and Ali Montazeri. Health-related quality of life in breast cancer patients: review of reviews from 2008 to 2018. *Health and quality of life outcomes*, 18:1–25, 2020.

[23] Md Bokhtiar Hasan, Masnun Mahi, Tapan Sarker, and Md Ruhul Amin. Spillovers of the covid-19 pandemic: Impact on global economic activity, the stock market, and the energy sector. *Journal of Risk and Financial Management*, 14(5):200, 2021.

[24] Aaron Hawley. Radio's influence on music from 1919 to 1926. 2000.

[25] Abhishek Pratap, Elias Chaibub Neto, Phil Snyder, Carl Stepnowsky, Noémie Elhadad, Daniel Grant, Matthew H Mohebbi, Sean Mooney, Christine Suver, John Wilbanks, et al. Indicators of retention in remote digital health studies: a cross-study evaluation of 100,000 participants. *NPJ digital medicine*, 3(1):21, 2020.

[26] Matea Pavic, Vanessa Klaas, Gudrun Theile, Johannes Kraft, Gerhard Tröster, David Blum, and Matthias Guckenberger. Mobile health technologies for continuous monitoring of cancer patients in palliative care aiming to predict health status deterioration: a feasibility study. *Journal of palliative medicine*, 23(5):678–685, 2020.

[27] Idar Johan Brekke, Lars Håland Puntervoll, Peter Bank Pedersen, John Kellett, and Mikkel Brabrand. The value of vital sign trends in predicting and monitoring clinical deterioration: A systematic review. *PloS one*, 14(1):e0210875, 2019.

[28] Woori Bae, Kyunghoon Kim, and Bongjin Lee. Distribution of pediatric vital signs in the emergency department: a nationwide study. *Children*, 7(8):89, 2020.

[29] James S Shahoud, Terrence Sanvictores, and Narothama R Aeddula. Physiology, arterial pressure regulation. 2019.

[30] G Smith, O Redfern, M Pimentel, Stephen Gerry, G Collins, James Malycha, David Prytherch, P Schmidt, and P Watkinson. The national early warning score 2 (news2). *Clinical Medicine*, 19(3), 2019.

[31] Marco AF Pimentel, Oliver C Redfern, Stephen Gerry, Gary S Collins, James Malycha, David Prytherch, Paul E Schmidt, Gary B Smith, and Peter J Watkinson. A comparison of the ability of the national early warning score and the national early warning score 2 to identify patients at risk of in-hospital mortality: a multi-centre database study. *Resuscitation*, 134:147–156, 2019.

[32] Elizabeth M King, Hannah L Randolph, Maria S Floro, and Jooyeoun Suh. Demographic, health, and economic transitions and the future care burden. *World Development*, 140:105371, 2021.

[33] Chiara Achangwa, Huikyung Park, Sukhyun Ryu, and Moo-Sik Lee. Collateral impact of public health and social measures on respiratory virus activity during the covid-19 pandemic 2020–2021. *Viruses*, 14(5):1071, 2022.

[34] Chronic diseases. https://www.cdc.gov/chronicdisease/resources/infographic/chronic-diseases.htm, 2023. Last accessed on 20/03/2023.

[35] U.S. health care spending. https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/nationalhealthexpenddata/nationalhealthaccountshistorical#:~:text=The%20data%20are%20presented%20by,spending%20accounted%20for%2018.3%20percent, 2022. last Accede 20/03/2023.

[36] Stephen Bryant, David Evans, Andy Hopkins, and Jocelyn Potter. *The story of the Internet*. Pearson Education, 2008.

[37] Kevin Ashton et al. That 'internet of things' thing. *RFID journal*, 22(7):97–114, 2009.

[38] Roberto Minerva, Abyi Biru, and Domenico Rotondi. Towards a definition of the internet of things (iot). *IEEE Internet Initiative*, 1(1):1–86, 2015.

[39] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[40] Iot definition. https://www.oracle.com/internet-of-things/what-is-iot/, 2023. Last accessed on 20/03/2023.

[41] The internet of things at the ietf. https://www.ietf.org/topics/iot/, 2022. Last accessed on 20/03/2023.

[42] Perry Xiao. *Designing Embedded Systems and the Internet of Things (IoT) with the ARM mbed*. John Wiley & Sons, 2018.

[43] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[44] Pallavi Sethi and Smruti R Sarangi. Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017, 2017.

[45] Shadi Al-Sarawi, Mohammed Anbar, Kamal Alieyan, and Mahmood Alzubaidi. Internet of things (iot) communication protocols. In *2017 8th International conference on information technology (ICIT)*, pages 685–690. IEEE, 2017.

[46] Mahdi Fahmideh and Didar Zowghi. An exploration of iot platform development. *Information Systems*, 87:101409, 2020.

[47] Ahmed A Ismail, Haitham S Hamza, and Amira M Kotb. Performance evaluation of open source iot platforms. In *2018 IEEE global conference on internet of things (GCIoT)*, pages 1–5. IEEE, 2018.

[48] P Manojkumar, M Suresh, Alim Al Ayub Ahmed, Hitesh Panchal, Christopher Asir Rajan, A Dheepanchakkravarthy, A Geetha, B Gunapriya, Suman Mann, and Kishor Kumar Sadasivuni. A novel home automation distributed server management system using internet of things. *International Journal of Ambient Energy*, 43(1):5478–5483, 2022.

[49] Saiteja Prasad Chatrati, Gahangir Hossain, Ayush Goyal, Anupama Bhan, Sayantan Bhattacharya, Devottam Gaurav, and Sanju Mishra Tiwari. Smart home health monitoring system for predicting type 2 diabetes and hypertension. *Journal of King Saud University-Computer and Information Sciences*, 34(3):862–870, 2022.

[50] Rose Mary John, Finky Francis, Joe Neelankavil, Alwyn Antony, Ancy Devassy, and KJ Jinesh. Smart public transport system. In *2014 International Conference on Embedded Systems (ICES)*, pages 166–170. IEEE, 2014.

[51] Ashutosh Sharma, Mikhail Georgi, Maxim Tregubenko, Alexey Tselykh, and Alexander Tselykh. Enabling smart agriculture by implementing artificial intelligence and embedded sensing. *Computers & Industrial Engineering*, 165:107936, 2022.

[52] Jing Zhang, Qichen Yan, Xiaogang Zhu, and Keping Yu. Smart industrial iot empowered crowd sensing for safety monitoring in coal mine. *Digital Communications and Networks*, 2022.

[53] Muhammad Saqib Jamil, Muhammad Atif Jamil, Anam Mazhar, Ahsan Ikram, Abdullah Ahmed, and Usman Munawar. Smart environment monitoring system by employing wireless sensor networks on vehicles for pollution free smart cities. *Procedia Engineering*, 107:480–484, 2015.

[54] Kwok Tai Chui, Wadee Alhalabi, Sally Shuk Han Pang, Patricia Ordóñez de Pablos, Ryan Wen Liu, and Mingbo Zhao. Disease diagnosis in smart healthcare: Innovation, technologies and applications. *Sustainability*, 9(12):2309, 2017.

[55] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.

[56] Lu Hou, Shaohang Zhao, Xiong Xiong, Kan Zheng, Periklis Chatzimisios, M Shamim Hossain, and Wei Xiang. Internet of things cloud: Architecture and implementation. *IEEE Communications Magazine*, 54(12):32–39, 2016.

[57] Mohammad Ilyas Malik, Shahid Hussain Wani, and Abreen Rashid. Cloud computing-technologies. *International Journal of Advanced Research in Computer Science*, 9(2), 2018.

[58] Maryam Ahmadi and Nasim Aslani. Capabilities and advantages of cloud computing in the implementation of electronic health record. *Acta Informatica Medica*, 26(1):24, 2018.

[59] Sadia Ijaz, Tauqeer Safdar, and Amanullah Khan. Challenges and limitation of resource allocation in cloud computing. In *International Conference on Intelligent Technologies and Applications*, pages 723–737. Springer, 2019.

[60] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16, 2012.

[61] Michaela Iorga, Larry Feldman, Robert Barton, Michael J Martin, Nedim S Goren, Charif Mahmoudi, et al. Fog computing conceptual model. 2018.

[62] Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. Fog computing: Platform and applications. In *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)*, pages 73–78. IEEE, 2015.

[63] Mohamed Firdhous, Osman Ghazali, and Suhaidi Hassan. Fog computing: Will it be the future of cloud computing? 2014.

[64] Yu Cao, Songqing Chen, Peng Hou, and Donald Brown. Fast: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In *2015 IEEE international conference on networking, architecture and storage (NAS)*, pages 2–11. IEEE, 2015.

[65] Sonia Singh, Ankita Bansal, Rajinder Sandhu, and Jagpreet Sidhu. Fog computing and iot based healthcare support service for dengue fever. *International Journal of Pervasive Computing and Communications*, 2018.

[66] Vladimir Stantchev, Ahmed Barnawi, Sarfaraz Ghulam, Johannes Schubert, and Gerrit Tamm. Smart items, fog and cloud computing as enablers of servitization in healthcare. *Sensors & Transducers*, 185(2):121–128, 2014.

[67] Ubaid Ur Rehman, Seong-Bae Park, and Sungyoung Lee. Secure health fog: A novel framework for personalized recommendations based on adaptive model tuning. *IEEE Access*, 9:108373–108391, 2021.

[68] Fadi Al-Turjman and F Al-Turjman. *Edge computing*. Springer, 2019.

[69] Gonçalo Carvalho, Bruno Cabral, Vasco Pereira, and Jorge Bernardino. Edge computing: current trends, research challenges and future directions. *Computing*, 103(5):993–1023, 2021.

[70] Peiran Dong, Zhaolong Ning, Mohammad S Obaidat, Xin Jiang, Yi Guo, Xiping Hu, Bin Hu, and Balqies Sadoun. Edge computing based healthcare systems: Enabling decentralized health monitoring in internet of medical things. *IEEE Network*, 34(5):254–261, 2020.

[71] Ghulam Muhammad, Mohammed F Alhamid, Mansour Alsulaiman, and Brij Gupta. Edge computing with cloud for voice disorder assessment and treatment. *IEEE Communications Magazine*, 56(4):60–65, 2018.

[72] Muhammad Usman Ashraf, Abdul Hannan, Sehrish Munawar Cheema, Zahra Ali, Abdulraheem Alofi, et al. Detection and tracking contagion using iot-edge technologies: Confronting covid-19 pandemic. In *2020 international conference on electrical, communication, and computer engineering (ICECCE)*, pages 1–6. IEEE, 2020.

[73] Medical device. https://www.marketsandmarkets.com/Market-Reports/iot-medical-device-market-15629287.html, 2023. Last accessed on 20/03/2023.

[74] health IoT device installations in 2020. https://www.statista.com/statistics/735810/healthcare-iot-installations-global-estimate/, 2023. Last accessed on 20/03/2023".

[75] Pooja Gupta and MA Alam. Challenges in the adaptation of iot technology. *A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems*, pages 347–369, 2022.

[76] Panagiotis I Radoglou Grammatikis, Panagiotis G Sarigiannidis, and Ioannis D Moscholios. Securing the internet of things: Challenges, threats and solutions. *Internet of Things*, 5:41–70, 2019.

[77] Murtaza Ahmed Siddiqi, Heejung Yu, and Jingon Joung. 5g ultra-reliable low-latency communication implementation challenges and operational issues with iot devices. *Electronics*, 8(9):981, 2019.

[78] AR Al-Ali, Ahmad Al Nabulsi, Shayok Mukhopadhyay, Mohammad Shihab Awal, Sheehan Fernandes, and Khalil Ailabouni. Iot-solar energy powered smart farm irrigation system. *Journal of Electronic Science and Technology*, 17(4):100017, 2019.

[79] Interoperability meaning. https://dictionary.cambridge.org/dictionary/english/interoperability, 2023. Last accessed on 20/03/2023.

[80] Institute of Electrical and Electronics Engineers. Ieee standard computer dictionary: A compilation of ieee standard computer glossaries. *IEEE Std 610*, pages 1–217, 1991.

[81] Hafizur Rahman and Md Iftekhar Hussain. A comprehensive survey on semantic interoperability for internet of things: State-of-the-art and research challenges. *Transactions on Emerging Telecommunications Technologies*, 31(12):e3902, 2020.

[82] Antonios Pliatsios, Christos Goumopoulos, and Konstantinos Kotis. A review on iot frameworks supporting multi-level interoperability—the semantic social network of things framework. *Int. J. Adv. Internet Technol*, 13(1):46–64, 2020.

[83] Mahda Noura, Mohammed Atiquzzaman, and Martin Gaedke. Interoperability in internet of things: Taxonomies and open challenges. *Mobile networks and applications*, 24:796–809, 2019.

[84] Preeti Agarwal and Mansaf Alam. Open service platforms for iot. *Internet of Things (IoT) Concepts and Applications*, pages 43–59, 2020.

[85] Semantic interoperability. https://www.w3.org/2001/sw/BestPractices/OEP/SemInt/, 2005. Last accessed on 20/03/2023.

[86] Dominique Guinard, Vlad Trifa, Friedemann Mattern, and Erik Wilde. From the internet of things to the web of things: Resource-oriented architecture and best practices. *Architecting the Internet of things*, pages 97–129, 2011.

[87] Ege Korkan, Sebastian Kaebisch, Matthias Kovatsch, and Sebastian Steinhorst. Safe interoperability for web of things devices and systems. *Languages, Design Methods, and Tools for Electronic System Design: Selected Contributions from FDL 2018*, pages 47–69, 2020.

[88] RDF. https://www.w3.org/RDF/. Accessed on February 23, 2023.

[89] RDF schema. https://www.w3.org/TR/rdf-schema/. Accessed on avril 23, 2023.

[90] web ontology language (owl). https://www.w3.org/OWL/. Accessed on avril 23, 2023.

[91] Vijayan Sugumaran and Veda C Storey. The role of domain ontologies in database design: An ontology management and conceptual modeling environment. *ACM Transactions on Database Systems (TODS)*, 31(3):1064–1094, 2006.

[92] Lambertus Marie De Rijk. *Aristotle: Semantics and Ontology: Volume I: General Introduction. The Works on Logic*, volume 91. Brill, 2016.

[93] Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R Swartout. Enabling technology for knowledge sharing. *AI magazine*, 12(3):36–36, 1991.

[94] Tom Gruber. What is an ontology, 1993.

[95] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

[96] Nicola Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

[97] Patrick Lambrix and He Tan. Ontology alignment and merging. *Anatomy Ontologies for Bioinformatics: Principles and Practice*, pages 133–149, 2008.

[98] York Sure, Steffen Staab, and Rudi Studer. On-to-knowledge methodology (otkm). *Handbook on ontologies*, pages 117–132, 2004.

[99] Helena Sofia Pinto, Steffen Staab, and Christoph Tempich. Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In *ECAI*, volume 16, page 393, 2004.

[100] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics as ontology languages for the semantic web. *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, pages 228–248, 2005.

[101] Bettina Hoser, Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Semantic network analysis of ontologies. In *The Semantic Web: Research and Applications: 3rd European Semantic Web Conference, ESWC 2006 Budva, Montenegro, June 11-14, 2006 Proceedings 3*, pages 514–529. Springer, 2006.

[102] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[103] Protégé software. https://protege.stanford.edu/, 2000. Last accessed on 20/03/2023.

[104] Moritz Weiten. *Ontostudio® as a ontology engineering environment*. Springer, 2009.

[105] York Sure, Juergen Angele, and Steffen Staab. Ontoedit: Multifaceted inferencing for ontology engineering. *Journal on Data Semantics I*, pages 128–152, 2003.

[106] Sparql 1.1 query language. https://www.w3.org/TR/sparql11-query/. Accessed on February 23, 2023.

[107] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21(79):1–31, 2004.

[108] Ravi Bhushan Mishra and Sandeep Kumar. Semantic web reasoners and languages. *Artificial Intelligence Review*, 35:339–368, 2011.

[109] John W Aldrich. Population characteristics and nomenclature of the hermit thrush. *Proceedings of the United States National Museum*, 1968.

[110] Dennis Pfisterer, Kay Romer, Daniel Bimschas, Oliver Kleine, Richard Mietz, Cuong Truong, Henning Hasemann, Alexander Kröller, Max Pagel, Manfred Hauswirth, et al. Spitfire: toward a semantic web of things. *IEEE Communications Magazine*, 49(11):40–48, 2011.

[111] Sasikanth Avancha, Chintan Patel, and Anupam Joshi. Ontology-driven adaptive sensor networks. *UMBC Student Collection*, 2004.

[112] David J Russomanno, Cartik R Kothari, and Omoju A Thomas. Building a sensor ontology: A practical approach leveraging iso and ogc models. In *IC-AI*, pages 637–643, 2005.

[113] Holger Neuhaus and Michael Compton. The semantic sensor network ontology. In *AGILE workshop on challenges in geospatial data harmonisation, Hannover, Germany*, pages 1–33, 2009.

[114] Michael Compton, Payam Barnaghi, Luis Bermudez, Raul Garcia-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. *Journal of Web Semantics*, 17:25–32, 2012.

[115] Xiang Wang, Xiaoming Zhang, and Mei Li. A survey on semantic sensor web: sensor ontology, mapping and query. *International Journal of u-and e-Service, Science and Technology*, 8(10):325–342, 2015.

[116] Rimel Bendadouche, Catherine Roussey, Gil De Sousa, Jean-Pierre Chanet, and Kun Mean Hou. Extension of the semantic sensor network ontology for wireless sensor networks: The stimulus-wsnnode-communication pattern. In *5th International Workshop on Semantic Sensor Networks in conjunction with the 11th International Semantic Web Conference (ISWC)*, pages 16–p, 2012.

[117] Maria Bermudez-Edo, Tarek Elsaleh, Payam Barnaghi, and Kerry Taylor. Iot-lite: a lightweight semantic model for the internet of things and its use with dynamic semantics. *Personal and Ubiquitous Computing*, 21:475–487, 2017.

[118] Krzysztof Janowicz, Armin Haller, Simon JD Cox, Danh Le Phuoc, and Maxime Lefrançois. Sosa: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, 56:1–10, 2019.

[119] Tarek Elsaleh, Shirin Enshaeifar, Roonak Rezvani, Sahr Thomas Acton, Valentinas Janeiko, and Maria Bermudez-Edo. Iot-stream: A lightweight ontology for internet of things data streams and its use with data analytics and event detection services. *Sensors*, 20(4):953, 2020.

[120] Amelie Gyrard and Martin Serrano. Fiesta-iot: Federated interoperable semantic internet of things (iot) testbeds and applications. In *ICT*, 2015.

[121] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Iot-o, a core-domain iot ontology to represent connected devices networks. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pages 561–576. Springer, 2016.

[122] Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, and Karima Boudaoud. Cross-domain internet of things application development: M3 framework and evaluation. In *2015 3rd International conference on future Internet of Things and Cloud*, pages 9–16. IEEE, 2015.

[123] Mahdi Ben Alaya, Samir Medjiah, Thierry Monteil, and Khalil Drira. Toward semantic interoperability in onem2m architecture. *IEEE Communications Magazine*, 53(12):35–41, 2015.

[124] Laura Daniele, Frank den Hartog, and Jasper Roes. Created in close interaction with the industry: the smart appliances reference (saref) ontology. In *Formal Ontologies Meet Industry: 7th International Workshop, FOMI 2015, Berlin, Germany, August 5, 2015, Proceedings 7*, pages 100–112. Springer, 2015.

[125] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[126] Lotfi A Zadeh. Knowledge representation in fuzzy logic. *An introduction to fuzzy logic applications in intelligent systems*, pages 1–25, 1992.

[127] Lotfi Asker Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3):199–249, 1975.

[128] Elie Sanchez. *Fuzzy logic and the semantic web*. Elsevier, 2006.

[129] Omar Adil M Ali, Aous Y Ali, and Balasem Salem Sumait. Comparison between the effects of different types of membership functions on fuzzy logic controller performance. *International Journal*, 76:76–83, 2015.

[130] EH Mamdani and S Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of human-computer studies*, 51(2):135–147, 1999.

[131] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15:116–132, 1985.

[132] Tofigh Allahviranloo and Rahim Saneifard. Defuzzification method for ranking fuzzy numbers based on center of gravity. 2012.

[133] Zongmin Ma, Fu Zhang, Li Yan, and Jingwei Cheng. *Fuzzy knowledge management for the semantic web*. Springer, 2014.

[134] Hanen Ghorbel, Afef Bahri, and Rafik Bouaziz. Fuzzy ontologies building method: Fuzzy ontomethodology. In *2010 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–8. IEEE, 2010.

[135] Firas Zekri, Emna Turki, and Rafik Bouaziz. Alzfuzzyonto: Une ontologie floue pour l'aide à la décision dans le domaine de la maladie d'alzheimer. In *INFORSID*, pages 83–98, 2015.

[136] Fernando Bobillo and Umberto Straccia. The fuzzy ontology reasoner fuzzydl. *Knowledge-Based Systems*, 95:12–34, 2016.

[137] Silvia Calegari and Davide Ciucci. Fuzzy ontology and fuzzy-owl in the kaon project. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6. IEEE, 2007.

[138] Fernando Bobillo and Umberto Straccia. Fuzzy ontology representation using owl 2. *International journal of approximate reasoning*, 52(7):1073–1094, 2011.

[139] Representation using OWL 2 plugin. http://www.umbertostraccia.it/cs/software/FuzzyOWL/, 2019. Last accessed on 20/03/2023.

[140] Mark A Beyer and Douglas Laney. he Importance of Big Data: A Definition. https://www.gartner.com/en/documents/2057415, 2012. Last accessed on 20/03/2023.

[141] Big data Definition. https://www.oracle.com/big-data/what-is-big-data/., 2023. Last accessed on 20/03/2023.

[142] Wo L Chang and Nancy Grady. Nist big data interoperability framework: Volume 1, definitions. 2019.

[143] Paul Zikopoulos and Chris Eaton. *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.

[144] Richard K Lomotey and Ralph Deters. Towards knowledge discovery in big data. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pages 181–191. IEEE, 2014.

[145] Statista. Big data volume. https://www.statista.com/statistics/1185900/worldwide-datasphere-storage-capacity-installed-base/, 2022. Last accessed on 20/03/2023.

[146] Samiksha Shukla, K Balachandran, and VS Sumitha. A framework for smart transportation using big data. In *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, pages 1–3. IEEE, 2016.

[147] IoT dig data size. https://blogs.idc.com/2019/11/04/how-you-contribute-to-todays-growing-datasphere-and-its-enterprise-impact/, 2019. Last accessed on 20/03/2023.

[148] Big data MARKET. https://www.idc.com/getdoc.jsp?containerId=US48083022, 2022. Last accessed on 20/03/2023.

[149] Sulayman K Sowe, Takashi Kimata, Mianxiong Dong, and Koji Zettsu. Managing heterogeneous sensor data on a big data platform: Iot services for data-intensive science. In *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, pages 295–300. IEEE, 2014.

[150] David Ostrowski, Nestor Rychtyckyj, Perry MacNeille, and Mira Kim. Integration of big data using semantic web technologies. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 382–385. IEEE, 2016.

[151] Zaoui Sayah, Okba Kazar, Brahim Lejdel, Abdelkader Laouid, and Ahmed Ghenabzia. An intelligent system for energy management in smart cities based on big data and ontology. *Smart and Sustainable Built Environment*, 10(2):169–192, 2021.

[152] DBpedia. http://wikidata.dbpedia.org/develop/datasets/latest-core-dataset-releases. (accessed: 23.10.2022).

[153] Lubm. http://swat.cse.lehigh.edu/projects/lubm/. (accessed: 23.04.2023).

[154] lod-cloud. https://lod-cloud.net/. (accessed: 23.04.2023),.

[155] Li Ma, Zhong Su, Yue Pan, Li Zhang, and Tao Liu. Rstar: an rdf storage and query system for enterprise resource management. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 484–491, 2004.

[156] Daniel J Abadi, Adam Marcus, Samuel R Madden, and Kate Hollenbach. Sw-store: a vertically partitioned dbms for semantic web data management. *The VLDB Journal*, 18(2):385–406, 2009.

[157] Kevin Wilkinson and Kevin Wilkinson. Jena property table implementation, 2006.

[158] Li Yan, Ping Zhao, and Zongmin Ma. Indexing temporal rdf graph. *Computing*, 101(10):1457–1488, 2019.

[159] Apache-Jena. https://jena.apache.org/. (accessed: 23.04.2023).

[160] Graphdb. https://graphdb.ontotext.com. (accessed: 3.04.2023).

[161] lblazegraph. https://blazegraph.com/. (accessed: 23.04.2023),.

[162] Rdflib. https://rdflib.readthedocs.io/en/stable/#source-code. (accessed: 3.04.2022).

[163] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, 1984.

[164] Ibtissem Kemouguette, Zineddine Kouahla, Ala-Eddine Benrazek, Brahim Farou, and Hamid Seridi. Cost-effective space partitioning approach for iot data indexing and retrieval. In *2021 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–6. IEEE, 2021.

[165] Venkathachary Srinivasan and Michael J Carey. Performance of b+ tree concurrency control algorithms. *The VLDB Journal*, 2:361–406, 1993.

[166] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad hoc networks*, 10(7):1497–1516, 2012.

[167] Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. Semantics for the internet of things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(1):1–21, 2012.

[168] Sergios Soursos, Ivana Podnar Žarko, Patrick Zwickl, Ivan Gojmerac, Giuseppe Bianchi, and Gino Carrozzo. Towards the cross-domain interoperability of iot platforms. In *2016 European conference on networks and communications (EuCNC)*, pages 398–402. IEEE, 2016.

[169] Oscar Novo and Mario Di Francesco. Semantic interoperability in the iot: extending the web of things architecture. *ACM Transactions on Internet of Things*, 1(1):1–25, 2020.

[170] Stefan Poslad, Stuart E Middleton, Fernando Chaves, Ran Tao, Ocal Necmioglu, and Ulrich Bügel. A semantic iot early warning system for natural environment crisis management. *IEEE Transactions on Emerging Topics in Computing*, 3(2):246–257, 2015.

[171] Yu Hsiang Chien and Fuchun Joseph Lin. Distributed semantic reasoning enabled by fog computing. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1033–1040. IEEE, 2019.

[172] Nenad Petrovic and Milorad Tosic. Smada-fog: Semantic model driven approach to deployment and adaptivity in fog computing. *Simulation Modelling Practice and Theory*, 101:102033, 2020.

[173] Kristina Sahlmann, Thomas Scheffler, and Bettina Schnor. Ontology-driven device descriptions for iot network management. In *2018 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE, 2018.

[174] Dhananjay Singh, Gaurav Tripathi, Antonio M Alberti, and Antonio Jara. Semantic edge computing and iot architecture for military health services in battlefield. In *2017 14th IEEE annual consumer communications & networking conference (CCNC)*, pages 185–190. IEEE, 2017.

[175] Nidhi Pathak, Anandarup Mukherjee, and Sudip Misra. Sembox: Semantic interoperability in a box for wearable e-health devices. *IEEE Journal of Biomedical and Health Informatics*, 2022.

[176] Konstantin Ryabinin and Svetlana Chuprina. Ontology-driven edge computing. In *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part VII*, pages 312–325. Springer, 2020.

[177] Xiang Su, Pingjiang Li, Jukka Riekki, Xiaoli Liu, Jussi Kiljander, Juha-Pekka Soininen, Christian Prehofer, Huber Flores, and Yuhong Li. Distribution of semantic reasoning on the edge of internet of things. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9. IEEE, 2018.

[178] Michael Dibley, Haijiang Li, Yacine Rezgui, and John Miles. An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction*, 28:1–14, 2012.

[179] Joao Moreira, Luís Ferreira Pires, Marten van SINDEREN, and Laura Daniele. Saref4health: Iot standard-based ontology-driven healthcare systems. In *FOIS*, pages 239–252, 2018.

[180] Garvita Bajaj, Rachit Agarwal, and Pushpendra Singh. A study of existing ontologies in the iot-domain. *arXiv preprint arXiv:1707.00112*, 2017.

[181] Amelie Gyrard, Christian Bonnet, Karima Boudaoud, and Martin Serrano. Lov4iot: A second life for ontology-based domain knowledge to build semantic web of things applications. In *2016 IEEE 4th international conference on future internet of things and cloud (FiCloud)*, pages 254–261. IEEE, 2016.

[182] Ben YB Kim and Joon Lee. Smart devices for older adults managing chronic disease: a scoping review. *JMIR mHealth and uHealth*, 5(5):e7141, 2017.

[183] Ahlem Rhayem, Mohamed Ben Ahmed Mhiri, Mayssa Ben Salah, and Faiez Gargouri. Ontology-based system for patient monitoring with connected objects. *Procedia computer science*, 112:683–692, 2017.

[184] Roberto Reda, Filippo Piccinini, and Antonella Carbonaro. Towards consistent data representation in the iot healthcare landscape. In *Proceedings of the 2018 International Conference on Digital Health*, pages 5–10, 2018.

[185] Mnasser Houda, Maha Khemaja, Kathia Oliveira, and Mourad Abed. A public transportation ontology to support user travel planning. In *2010 Fourth International Conference on Research Challenges in Information Science (RCIS)*, pages 127–136. IEEE, 2010.

[186] AJ Bermejo, Jesus Villadangos, Jose Javier Astrain, Alberto Cordoba, Leire Azpilicueta, Uxue Garate, and Francisco Falcone. Ontology based road traffic management in emergency situations. *Adhoc & Sensor Wireless Networks*, 20, 2014.

[187] Filippo Benvenuti, Claudia Diamantini, Domenico Potena, and Emanuele Storti. An ontology-based framework to support performance monitoring in public transport systems. *Transportation Research Part C: Emerging Technologies*, 81:188–208, 2017.

[188] Michele Ruta, Floriano Scioscia, Eugenio Di Sciascio, and Domenico Rotondi. Ubiquitous knowledge bases for the semantic web of things. In *First Internet of Things International Forum*, 2011.

[189] Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, and Karima Boudaoud. Integrating machine-to-machine measurement framework into onem2m architecture. In *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 364–367. IEEE, 2015.

[190] Ivana Podnar Žarko, Szymon Mueller, Marcin Płociennik, Tomasz Rajtar, Michael Jacoby, Matteo Pardi, Gianluca Insolvibile, Vasileios Glykantzis, Aleksandar Antonić, Mario Kušek, et al. The symbiote solution for semantic and syntactic interoperability of cloud-based iot platforms. In *2019 Global IoT Summit (GIoTS)*, pages 1–6. IEEE, 2019.

[191] Andrea Cimmino, Viktor Oravec, Fernando Serena, Peter Kostelnik, María Poveda-Villalón, Athanasios Tryferidis, Raúl García-Castro, Stefan Vanya, Dimitrios Tzovaras, and Christoph Grimm. Vicinity: Iot semantic interoperability based on the web of things. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 241–247. IEEE, 2019.

[192] Paweł Szmeja. Generic ontology for iot platforms. https://inter-iot.github.io/ontology/, 2018. last accede 2023.04.25.

[193] Giancarlo Fortino, Claudio Savaglio, Carlos E Palau, Jara Suarez de Puga, Maria Ganzha, Marcin Paprzycki, Miguel Montesinos, Antonio Liotta, and Miguel Llop. Towards multi-layer interoperability of heterogeneous iot platforms: The inter-iot approach. *Integration, interconnection, and interoperability of IoT systems*, pages 199–232, 2018.

[194] Arne Bröring, Andreas Ziller, Victor Charpenay, Aparna S Thuluva, Darko Anicic, Stefan Schmid, Achille Zappa, Mari Paz Linares, Lars Mikkelsen, and Christian Seidel. The big iot api-semantically enabling iot interoperability. *IEEE Pervasive Computing*, 17(4):41–51, 2018.

[195] Natalia Díaz Rodríguez, Manuel P Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66:46–60, 2014.

[196] Farman Ali, Daehan Kwak, Pervez Khan, SM Riazul Islam, Kye Hyun Kim, and Kyung Suk Kwak. Fuzzy ontology-based sentiment analysis of transportation and city feature reviews for safe traveling. *Transportation Research Part C: Emerging Technologies*, 77:33–48, 2017.

[197] Olaide N Oyelade, Absalom E Ezugwu, and Sunday A Adewuyi. Enhancing reasoning through reduction of vagueness using fuzzy owl-2 for representation of breast cancer ontologies. *Neural Computing and Applications*, 34(4):3053–3078, 2022.

[198] Nora Shoaip, Amira Rezk, Shaker El-Sappagh, Louai Alarabi, Sherif Barakat, and Mohammed M Elmogy. A comprehensive fuzzy ontology-based decision support system for alzheimer's disease diagnosis. *IEEE Access*, 9:31350–31372, 2020.

[199] Shaker El-Sappagh, José M Alonso, Farman Ali, Amjad Ali, Jun-Hyeog Jang, and Kyung-Sup Kwak. An ontology-based interpretable fuzzy decision support system for diabetes diagnosis. *IEEE Access*, 6:37371–37394, 2018.

[200] Kamil Bajda-Pawlikowski. Querying rdf data stored in dbms: Sparql to sql conversion. Technical report, Technical Report TR-1409, Yale Computer Science Department, USA, 2008.

[201] Hajung Sung, Suckhoon Lee, and Doo-Kwon Baik. Comparison evaluation for sparql-to-sql translation model. In *Proceedings of the International Conference on Semantic Web and Web Services (SWWS)*, page 52. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2013.

[202] Nitinder Mohan and Jussi Kangasharju. Edge-fog cloud: A distributed cloud for internet of things computations. In *2016 Cloudification of the Internet of Things (CIoT)*, pages 1–6. IEEE, 2016.

[203] Abdelhalim Hadjadj and Khaled Halimi. Improving health disabled people through smart wheelchair based on fuzzy ontology. In *2021 8th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 1–6. IEEE, 2021.

[204] Saref4wear ontology. `https://saref.etsi.org/saref4wear/v1.1.1/`, 2020. last accede 2023.04.25.

[205] Siba Kumar Udgata and Nagender Kumar Suryadevara. *Internet of Things and sensor network for COVID-19*. Springer, 2021.

[206] Who algeria. https://www.aps.dz/en/health-science-technology/34007-who-representative-praises-algeria-measures-to-stem-coronavirus-pandemic, 2020. last accede 2023.04.25.

[207] Shaker El-Sappagh, Mohammed Elmogy, and AM Riad. A fuzzy-ontology-oriented case-based reasoning framework for semantic diabetes diagnosis. *Artificial intelligence in medicine*, 65(3):179–208, 2015.

[208] Marius Myrstad, Håkon Ihle-Hansen, Anders Aune Tveita, Elizabeth Lyster Andersen, Ståle Nygård, Arnljot Tveit, and Trygve Berge. National early warning score 2 (news2) on admission predicts severe disease and in-hospital mortality from covid-19–a prospective cohort study. *Scandinavian journal of trauma, resuscitation and emergency medicine*, 28:1–8, 2020.

[209] bio2rdf. https://bio2rdf.org/. (accessed: 23.10.2022), key=bio2rdf2022.

[210] Shane B Eisenman, Emiliano Miluzzo, Nicholas D Lane, Ronald A Peterson, Gahng-Seop Ahn, and Andrew T Campbell. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)*, 6(1):1–39, 2010.

[211] Jodi A Mindell, Erin S Leichman, Jordana Composto, Christina Lee, Bula Bhullar, and Russel M Walters. Development of infant and toddler sleep patterns: real-world data from a mobile application. *Journal of sleep research*, 25(5):508–516, 2016.

[212] Jeremy J Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83, 2004.

[213] Pablo Cingolani and Jesus Alcala-Fdez. jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation. In *2012 IEEE International Conference on Fuzzy Systems*, pages 1–8. IEEE, 2012.

[214] Vital signs of volunteers. https://figshare.com/articles/dataset/Vital_signs_of_15_Volunteers/5562298/1, 2017. last accede 2023.04.25.

[215] Pugazhvannan Cr, Ilavarasi Vanidassane, Dhivya Pownraj, Ravichandran Kandasamy, and Aneesh Basheer. National early warning score 2 (news2) to predict poor outcome in hospitalised covid-19 patients in india. *Plos one*, 16(12):e0261376, 2021.

[216] Mimic. https://physionet.org/content/mimiciii/1.4/. Accessed on February 23, 2023.

[217] BIDMC. https://archive.physionet.org/physiobank/database/bidmc/. Accessed on February 23, 2023.

AUTHOR'S PUBLICATION

International publications

1. Abdelhalim Hadjadj and Khaled Halimi. An integration of health monitoring system in public transport using the semantic web of things. JUCS: Journal of Universal Computer Science, 27(12), 2021.

2. Abdelhalim Hadjadj and Khaled Halimi. Covid-19 patients' health monitoring system using fuzzy ontology and internet of things. Iraqi Journal For Computer Science and Mathematics, 4(1):191–203, 2023.

International Communications

1. Abdelhalim Hadjadj and Khaled Halimi. Improving health disabled people through smart wheelchair based on fuzzy ontology. In 2021 8th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), pages 1–6. IEEE, 2021.

2. Hadjadj abdelhalim, halimi khaled, "the state of arte ontological for semantic interoperability in iot"; 3rd conference held in virtual mode on informatics and applied mathematic(IAM), organized by labstic laboratory, on october 21st-22nd, 2020.

3. Hadjadj abdelhalim, halimi khaled, "health monitoring system for the elderly using semantic web of things "; 4rd conference held in virtual mode on informatics and applied mathematics(IAM), organized by labstic laboratory,2021.

4. Hadjadj abdelhalim, halimi khaled, and zineddine kouahla, "Health Monitoring System for Older People based on Internet of Things using Fuzzy Ontology"; for an oral presentation in the 5th conference on informatics and applied mathematics(IAM), organized by labstic laboratory, 2022.