DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CREATING INTELLIGENT MANAGEMENT SYSTEM FOR TURKEY ON ONTOLOGICAL SENSOR DATA

by
Mehmet MİLLİ

February, 2021
İZMİR

CREATING INTELLIGENT MANAGEMENT SYSTEM FOR TURKEY ON ONTOLOGICAL SENSOR DATA

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering

by
Mehmet MİLLİ

February, 2021
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"CREATING INTELLIGENT MANAGEMENT SYSTEM FOR TURKEY ON ONTOLOGICAL SENSOR DATA"** completed by **MEHMET MİLLİ** under the supervision of **ASSIST. PROF. DR. ÖZLEM AKTAŞ** and we certify that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Assist. Prof. Dr. Özlem AKTAŞ

Supervisor

Assist. Prof. Dr. Kökten Ulaş BİRANT

Thesis Committee Member

Assist. Prof. Dr. Özgür TAMER

Thesis Committee Member

Assoc. Prof. Dr. Aytuğ ONAN

Examining Committee Member

Assoc. Prof. Dr. Seda POSTALCIOĞLU

Examining Committee Member

Prof.Dr. Özgür ÖZÇELİK

Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my supervisor, Assist. Prof. Özlem AKTAŞ, for the great mentorship of her immense knowledge, continuous support, supervision, criticism, and useful suggestions throughout this study. It was a great honor to work with her on this thesis together.

I also want to thank my thesis committee members, Assist. Prof. Kökten Ulaş BİRANT and Assist. Prof. Özgür TAMER, for their moral support and invaluable comments throughout this study.

I special thanks also to my twin Musa MİLLİ, my project partner Dr. Sanaz LAKESTANI and Dr. Bahram SARKARATI, my collogue İsmail Hakkı PARLAK and Dr. Emre ÜNSAL for their help and cooperation in developing the study.

Also, I would like to offer my special thanks to my family, my spiritual sister Aslı YILDIRIM, my son's spiritual family Kemal-Ayşe ERTÜRK, and their family for supporting me spiritually throughout writing this thesis and my life in general. I would not have been able to complete this thesis without their support and help.

Finally, I would like to express my special thanks to my family, my son Yavuz Selim MİLLİ (YSM) who is my motivation source, my wife Nursel MİLLİ for his patience and continuous support throughout this study.

Mehmet MİLLİ

CREATING INTELLIGENT MANAGEMENT SYSTEM FOR TURKEY ON ONTOLOGICAL SENSOR DATA

ABSTRACT

In recent years, sensors have become smaller enough to be used in every system, positive developments in the academic environment, and a decrease in prices have increased the interest in sensors. Sensor-based systems have spread rapidly to all areas of daily life, especially in industrial areas. Massive amounts of raw sensor data from sensor-based systems, the area of use of which has increased considerably, pose a fundamentally new set of research challenges, including their structuring, sharing, and management in a common framework. Although there are many academic studies on the integration of sensor data between different sensor-based systems, these studies focused on the integration of the data as syntax rather than semantic integration.

Nowadays, the semantic sensor web approach, which enables us to enrich the meaning of sensor data in order to provide more advanced access to sensor data and add annotations, has been seen by some researchers as a critical technology in solving these problems. The grand goal of this thesis is to provide a standard data model for heterogeneous sensor data from different platforms by extending the ontology of semantic sensor networks. The proposed system was tested using 8 indoor parameters collected in the Application and Research Center and Intensive Care Unit within Abant Izzet Baysal University. Sensor data collected from selected use-cases were added to the proposed framework and an RDF data set was created. Classic machine learning algorithms have been implemented on the RDF data set created and compared from different angles.

Keywords: Semantic sensor network, ontology modeling, heterogeneous sensor data, machine learning, stream data, real-time monitoring, data mining.

ONTOLOJİK SENSÖR VERİLERİ ÜZERİNE TÜRKİYE İÇİN AKILLI YÖNETİM SİSTEMİ OLUŞTURMA

ÖZ

Son yıllarda, sensorların her sistemde kullanılabilecek kadar küçülmeleri, akademik ortamdaki olumlu gelişmeler ve fiyatların düşmesi sonucu sensorlara duyulan ilgiyi arttırmıştır. Sensor tabanlı sistemler Endüstriyel alanlar başta olmak üzere günlük yaşamın her alanına hızla yayılmıştır. Kullanım alanı önemli ölçüde artan sensor tabanlı sistemlerden elde edilen çok fazla miktarda ham sensor verisi, ortak bir çerçevede yapılandırılması, paylaşılması ve yönetilmesi de dahil olmak üzere temelde yeni bir dizi araştırma zorlukları ortaya çıkarmaktadır. Sensor verilerinin farklı algılayıcı tabanlı sistemler arasında entegrasyonu konusunda bugüne kadar pek çok akademik çalışma bulunsa da bu çalışmalar genel olarak verilerin anlamsal entegrasyonu yerine sözdizimi olarak entegrasyonuna odaklanmıştır.

Günümüzde, sensor verilerine daha gelişmiş erişim sağlamak ve ek açıklamalar eklemek için sensor verilerinin anlamını zenginleştirmemizi sağlayan anlamsal sensor web yaklaşımı, bazı araştırmacılar tarafından bu sorunların çözümünde kritik bir teknoloji olarak görülmüştür. Bu tezin en büyük amacı anlamsal sensor ontolojisini genişleterek farklı platformlardan gelen heterojen sensor verileri için standart bir veri modeli sağlamaktır. Önerilen sistem Abant İzzet Baysal Üniversitesi bünyesinde bulunan Uygulama Merkezi ve Yoğun Bakım Ünitesinde toplanan sensor verileri ile test edilmiştir. Seçilen kullanım durumlarında toplanan sensor verileri önerilen çatıya eklenmiş ve RDF veri seti oluşturulmuştur. Oluşturulan RDF veri seti üzerinde klasik makine öğrenmesi algoritmaları entegre edilmiş ve farklı açılardan karşılaştırılmıştır.

Anahtar kelimeler: Anlamsal sensor ağı, ontoloji modelleme, heterojen sensor verileri, makine öğrenmesi, akış verileri, gerçek zamanlı izleme, veri madenciliği.

CONTENTS **Page**

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Overview

Wireless Sensor Networks (WSNs) are self-configurable systems to sensing phenomena in their deployment environment and to transmit data collected from different levels of a network to a point where data can be processed and analyzed (Radhika & Rangarajan, 2019). The general purpose of these networks is to observe the environment by sensing mechanical, thermal, biological, chemical, optical, and other phonemes in the real-world (Hussain, Cebi, & Shah, 2008). WSN's have been integrated into many applications since emerging and have become indispensable parts of many systems used in the industry (Tubaishat & Madria, 2003).

These networks were used in restricted areas in long years, such as surveillance and detection of nuclear, biological, and chemical attacks in the military, tracking of vehicles in the logistics area, due to some restrictions and their prices. Besides, they have been used partially in dangerous and inaccessible areas such as wildlife monitoring, volcanic eruption surveillance, etc.

WSNs have gained great more attention thanks to the developing wireless communication technology, the getting more cheapness the price of microprocessors, developments in energy supply systems, and sensor sizes becoming ideal for almost any application for several decades. They became more attracted in academia and industrial areas, due to their widespread nature and their wide deployment especially in the IoT, healthcare application, and other emerging fields (Karim & Zeadally, 2016).

1.2 Problem Definition

WSNs provide many advantages over traditional methods in terms of self-organization, fast transmission, flexibility, and secure data transmission. The ability to respond quickly to real-time events with action plans is one of the strengths of these

networks (Gungor, Hancke, & Member, 2009). However, the WSN study field has some constraints and problems, such as security, effective routing protocols, energy consumption, limited lifespan, and equipment costs. Moreover, one biggest challenge of WSN is that they are able to collect huge amounts of raw sensor data where they are deployed.

The raw data obtained from the sensors are frequently used in informatics and industrial fields. Sensor data is the output of a device that detects and responds to various phonemes in its physical environment. Generally, this output is used to provide information or provide input to another system. The use of sensors in many areas in our daily life has caused an exponential increase in the data obtained from the sensors. Such an excessive increase in sensor data makes it difficult to store and interpret the data (Aktaş, Milli, Lakestani, & Milli, 2020). Also, the lack of neither syntactic nor semantic integrity between these sensor data limits their sharing and reusability (Henson, Neuhaus, & Sheth, 2009). These inabilities can cause some problems with interoperability between disparate sensor networks that may have subtle variations in their sensing methods. To address these issues, the studies of the representation of sensor data, standardization of sensor data, and storage of sensor data have gained speed in recent years worldwide.

In this field, how to stored and interpreted when required this raw data collected by sensor nodes is one of the biggest problems to be solved in academia in recent years. Moreover, these systems also suffer from problems caused by the Internet environment, since these systems have recently become part of the Internet and information technologies with IoT studies. Sensors and the WSNs to which they are part are generally application-specific and cannot share sensor data with other applications, because, data from sensor networks with different operating principles are heterogeneous by their nature. Besides, since they are not reusable, they become unnecessary data after a certain time. The lack of specific standards of these raw sensor data makes it difficult to manage them.

The fact that the data received from the sensing networks are such heterogeneous and that they do not have a certain standard makes it difficult, to interpret and makes it impossible to reuse. Nowadays, some researchers state that the solution to this problem is the collation of the data associated with semantic web technologies (Barnaghi & Presser, 2010; Janowicz, Bröring, Stasch, & Everding, 2010; Mansour, Chbeir, & Arnould, 2019). The common point of their studies is that for the raw sensor data to be application-independent, required the meaning of the data is enriched to form a meaning pattern between each other. Therefore, a coherent infrastructure is needed to handle sensors of belong to different systems in an interoperable, platform-independent, and uniform way (Bröring et al., 2011). The SSW concept has been introduced to share, find, and access sensors and data in different applications.

Another common point of these studies that are given above is that data cannot be reused due to the lack of a certain standard among data collection applications. To overcome this problem, sensor data need specific standardization. There are two sensor data representation standards commonly used by researchers in the literature. These standards are the SWE developed by the Open Geospatial Consortium (OGC) (Percivall, Reed, & Davidson, 2007) and the Semantic Sensor Network (SSN) developed by the World Wide Web Consortium (W3C) (Compton et al., 2011).

1.3 Objectives of This Thesis and Contribution to Literature

The proposed Lightweight Ontological Framework for Heterogeneous Sensor data (OF4HeS:Lite) includes multiple objectives and scopes. These objectives may be separated into two categories as general and specific objectives. The objectives and scope of OF4HeS:Lite are as follows;

1.3.1 General Objectives

OF4HeS:Lite's main goal, is creating a sample ontology that has got a standard data model, using existing connections (classes, object properties, data properties, etc.) between environmental parameters (Temperature, Humidity, CO2, TVOC, PM2.5, PM10, CO, Light Level) and the SSN common data framework. Other objectives are

gains related to the collection of data using WSNs, ontological representation of data, and interpretation of them. These objectives are related to data science, data mining, machine learning, wireless communication, electromagnetic systems, and embedded systems. General achievements and outcomes expected at the end of this study from this data model are given following as items.

i. Since a common data model will be created for sensor data, heterogeneous sensor data collected from different systems would be managed on the same platform.
ii. The enormous amount of data from the sensors will be made more meaningful. Through this, the sustainability of sensor-based systems will be increased.
iii. Machine learning algorithms are more successful when structural data are given as input. Since proposed sensor ontology is based on a data model, when the ontological representation of sensor data is integrated into proactive systems that use machine learning algorithms, more meaningful inferences will be able to make.
iv. Since the sensor data is encoded with languages such as RDF and OWL, machine to machine (M2M) communication will be provided.
v. The proposed OF4HeS:Lite is a low-level sensor ontology. It is thought that OF4HeS:Lite will guide mid-level and high-level ontologies planned to be done next.
vi. To ensure that the collected sensor data is shared on the internet in appropriate formats such as CVS, SQL to enable it to be used by other researchers.

1.3.2 Specific Objectives

The specific objectives of OF4HeS:Lite includes the use-case environments where the proposed ontology is implemented. The proposed sensor ontology has been implemented on two different platforms to prove that sensor data in many different environments can be managed from a common system. The real-world use cases selected for this study are (i) Scientific research laboratories and (ii) Medical intensive care unit. Specific objectives and outcomes in terms of the use-case context expected at the end of this study from this data model are given following.

1.3.2.1 Specific Objectives in The Laboratory Environment

Bolu Abant Izzet Baysal University (BAIBU), Scientific Industrial Technological Application and Research Center (SITARC) has been chosen as the first use case for the proposed sensor ontology to be implemented. MALDI-TOF, AoxMercury, and Chromatography laboratories actively used in SITARC were selected as the measurement environment. In these laboratories, microorganism identification, proteomics analysis, bacterial count, fatty acid analysis, determination of anion-cation, total halogen determination, solid-phase extraction, etc. analyses are frequently performed.

In this case study, eight parameters in these laboratories were measured using five sensors. Keeping these parameters at the proper levels is extremely important for both the analysis to give healthy results. It is also important in terms of the health of the staff performing the analysis. The parameters of the laboratory environment previously obtained by passive sampling will be monitored in real-time with this proposed study. Following benefits will be ensured with this real-time monitoring;

i. When the parameters affected the result of laboratory analysis reach misleading levels, and reach unhealthy levels for the analyst, it will be detected, and appropriate action plans will be realized.
ii. As this real-time system replaces passive sampling, time, labor, and cost will be saved.
iii. Moreover, while the increase of some parameters is positive for human health, it affects the analysis results negatively, or vice-versa. Therefore, monitoring of laboratory environment parameters becomes more complicated. Thanks to the ontological rules created in OF4HeS:Lite, these complex situations are planned to be overcome.

1.3.2.2 Specific Objectives in The Intensive Care Unit

BAIBU Medical School Hospital Intensive Care Unit (MICU) has been chosen as the second use case for the proposed sensor ontology to be implemented. The proposed system has been established and evaluated in BAIBU MICU as a real-time surveillance system. Following benefits will be ensured with this real-time surveillance system;

iv. Providing real-time tracking of IAQ parameters in critical environments such as hospitals, and intensive care units.
v. To ensure that the relevant personnel is automatically informed when the air quality level of MICU decreases.
vi. Facilitating the interpretation of sensor data by hospital staff using data visualization software tools.
vii. To increase the awareness of hospital administrators about the importance of their investments in improving the politics of indoor air quality management systems.
viii. To establish a reliable, low cost, controllable, sustainable, computer-based indoor air quality management system prototype for hospital administrators.

1.4 Thesis Organization

This chapter, the overview, problem definition, and objectives of the thesis and the contribution to the field are stated. The general aim of this section presents a summary of exactly why this work was done and the motivation of the thesis. The rest of the thesis is given in the following paragraphs.

General information about three important areas used in this study is given in Chapter 2 of this thesis. These areas are WSNs, Semantic web technologies, and machine learning. This section aims to conduct a detailed examination of these academic fields, which have been researched, discussed and subject to this thesis for a few last decades.

Chapter 3 of this thesis presents a literature survey on WSNs and their challenges, SSN, sensor ontology, and machine learning-based prediction algorithms. In addition to this aim, this section discusses how other researchers represent sensor data with SW and ontologies, the motivation source of this study.

The methods and materials are explained in Chapter 4 of this thesis. This chapter includes the following topics: (i) Sensor selections used to collect data, (ii) Creation of node designs, (iii) Ontology framework used for modeling proposed sensor ontology, (iv) Development environments selection used in the study, (v) Selection of the triple store for proposed sensor ontology individuals and triples.

Chapter 5 of the thesis presents, how the proposed sensor ontology is applied to real-world usage situations. This chapter includes the following topics: (i) Introducing the use-case environments where the study will be implemented, (ii) Deployment of sensor nodes, (iii) Gathering of data, the creation of appropriate datasets. The main purpose of this section is to clearly show how the proposed sensor ontology is implemented in real-world use cases.

In Chapter 6 of the thesis, the comparison of the results of ML methods performed on the created data sets and the detection of the most appropriate algorithm on sensor ontology data from a variety of perspectives is explained in detail.

Finally, conclusions and discussion of the findings obtained in previous chapters are presented in Chapter 7. Also, this chapter contains future directions of the thesis, recommendations for more efficient sensor ontology in after studies.

CHAPTER TWO
BACKGROUND

2.1 Overview

In this chapter, the topics that will prepare the infrastructure for the proposed study will be discussed in detail. Firstly WSNs, their application areas of them, and their many issues will be handled. In addition, the place of SWTs and ontologies, among nowadays technologies, the final point that they come and their advantages will be presented. Finally, machine learning methods and approaches that would be considered to be used on the meaningful sensor data will be discussed to implement an effective proactive system.

2.2 Wireless Sensor Networks (WSNs)

WSNs are Micro-Electro-Mechanical Systems (MEMS) that are distributed to any medium for a specific purpose, are in constant communication with each other, can detect and measure the environment in which they are located. (Geylani, 2018). The concept of WSNs is a technology that first appeared in the 1980s and started to become very popular with the developments in wireless communication, usage areas of them have become quite widespread since they come into first use (Wang, 2010).

Generally, WSNs can contain hundreds or thousands of sensor nodes. Each sensor node in the WSN is capable of transmitting data to each other through the base station or Gateway Node and sends the data they collect directly or indirectly (hopping) to the collector center. (Ceyhan & Sağiroğlu, 2013). The sensor nodes may be wirelessly deployed randomized or in a certain order, into the environment in which they will collect the data. In a nutshell, WSNs consist of low-power, low-cost, and multifunctional sensor nodes with limited microprocessors and memory capacities, which can communicate over a short distance wireless environment (Sezer, Dogdu, & Ozbayoglu, 2018).

.

2.2.1 Basic Components of Wireless Sensor Networks

Sensor nodes in WSNs generally have a processing unit that manages the node, a communication unit that connects them to the network, and enables them to communicate with other nodes, a power unit that feeds the nodes, and different sensors according to the desired parameter in the environment (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002). In some applications, it can be added to units such as a power generator to regenerate the power unit, external memory when more memory is needed, and an actuation unit when the sensor node is asked to perform an action. The basic components of the nodes are given in Figure 2.1.

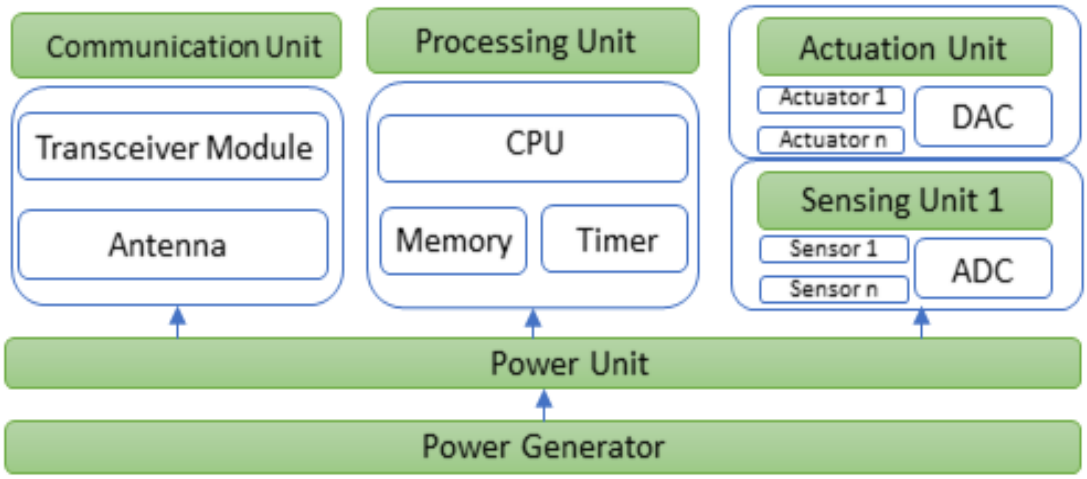


Figure 2.1 The main components of the sensor nodes

Today, WSNs have been integrated a wide variety of different implementation fields for existing and possible applications, thanks to their reliability, self-regulation, flexibility, and ease of installation (Zhao, Guibas, & Guibas, 2004). However, the use of WSN continues to become widespread day by day due to the fact that they do not want any infrastructure in the environments where they are installed before installation, they can operate smoothly after installation, do not require additional maintenance have a wide range of application areas. (Cheffena, 2012).

2.2.2 Challenges of WSNs

Although WSNs are technologies that are used and needed in all areas of daily life, today they have still some problems that await addressing. Many researchers are looking for different solutions to these restrictions in order to provide better service for WSNs. Within the scope of this thesis, as explained in Chapter 1 Introduction, some solutions are offered on how to make meaning of and interpret a lot of data that is one of these restrictions. Some of the restrictions that limit WSNs' providing better service and sometimes cause critical problems are described as follows.

2.2.2.1 Safety and Privacy

As with all information technologies, the security of society and people should always be the number one priority, in WSNs. They are frequently used in important areas that require national security, such as the military and civil environment traffic on highways, automation at the factory production stage, and environmental monitoring in agriculture. In WSNs, due to the nature of the communication, there is the possibility of intrusion and modification of data packets (Özdağ, 2016). Possible leakage of information in the WSN can lead to undesirable consequences, such as improper use or misuse of information. Therefore, it is imperative to provide security in this area so that the network can safely route data in the network to avoid these threats (Rani & Kumar, 2017).

2.2.2.2 Harsh Environmental Conditions

WSNs are used in harsh environments such as underwater, underground, etc. to getting useful raw sensor information. Generally, they are deployed to remote areas where there is little human surveillance. Since WSN nodes use wireless mode for communication, they are relatively more resistant to wired systems for remote and harsh environments (Hu, Wang, & Wan, 2013). However, nodes are ultimately electronic materials and are affected by harsh environmental conditions such as rain, wind, temperature. Depending on where they are located, they can sometimes be the target of wild animals and insects. Therefore, the sensor nodes that make up the WSNs,

and each hardware of the node must have unique features that ensure it withstands adverse conditions.

2.2.2.3 Limited Sources

Sensor nodes are small devices that they have only a small amount of memory and storage space to process and execute code on them. (Othman & Maga, 2018). Many microprocessors used in sensor nodes today have insufficient processing capacities to perform complex operations. Considering these limitations, the embedded system software that will collect data, transmit it to a gateway, or make the necessary decision must be effective and very small. However, in recent sensor-based studies, it is seen that the responsibilities and duties of the sensor nodes have increased. In real-time applications, in order to react more quickly to a negative scenario, sensor nodes can be expected to perform complex analysis and make decisions. In these cases, it may be necessary to add external memory and external space to increase the capacity of the sensor nodes. (Engel & Koch, 2016; Salle, Idiart, & Villavicencio, 2016).

2.2.2.4 Energy Consumption and Limited Life Span

One of the biggest problems with WSNs is that energy resources are very limited. Due to the lack of relevant infrastructure in the environments in which they are located, in most cases, maintenance of the nodes in the WSN and energy regeneration processes are important problems for the network. One of the most important factors that determine the life span of the network in WSNs is energy consumption. Thanks to the efficient use of the energies of the nodes, the life of the network can be extended, enabling them to perform their duties continuously (Aliyev, 2019).

2.2.2.5 Huge Amount of Raw Sensor Data

The main purpose of sensor nodes is to collect data. The sensors periodically detect, process, and transmit data from the surrounding environment to the base station or gateway. Depending on the number of nodes in a WSN and the number of various sensors integrated on each node, incredible sizes of data can be generated from a single

sensor network in a short time (Boubiche, Boubiche, & Toral-cruz, 2018). In what common format and how to store this data produced by the sensor nodes is the main subject of this thesis study. The sensor data obtained up to now in previous studies included application-specific uses. The integration of them with each other was impossible due to the lack of a common framework between the data where different sensor-based applications were collected. In this thesis, how to create a common framework for raw sensor data in different platforms, how to solve this problem is discussed in detail.

Apart from these problems and limitations described above, WSN has other problems. These can be listed as follows; Time Synchronization Issues (Ratna & Hansdah, 2015), Effective Deployment Issues (Boubrima, Bechkit, & Rivano, 2017), Robustness Issues (T. Qiu, Member, Zhao, & Member, 2017), Calibration Issues (Zion & Messer, 2014), Quality of Service Issues (S. Kaur & Mir, 2015), Self-Management Issues (Das, Misra, Member, Wolfinger, & Obaidat, 2016; Elsayed, Elhoseny, Sabbeh, & Riad, 2018), Fault Tolerance Issues (Chouikhi, El, Ghamri-doudane, & Azouz, 2015; M. Kaur & Garg, 2016), and etc. (Sharma, Bansal, & Bansal, 2013). While these problems await urgent and effective solutions, they continue to be handled in different ways in different studies.

2.3 Semantic Web Technologies and Ontologies

In the last ten years, the web has gained great importance in people's lives with the ability to access it from every device from every location and to transfer the vital processes of daily life such as finance, marketing, and education to this platform. Data on the web has increased day by day due to the fact that the web has such a place in daily life. This extraordinary increase has made it difficult to reach the right information on the web using classical methods. In all this information confusion, machines have been only capable of delivering web content, and they had could not understand, interpret, or make logical inferences about them.

To address these challenges and to present the solution, the Semantic Web concept was raised by Tim Berners-Lee and his team in the late 1990s. According to Tim Berners-Lee's, the SW is an extension of existing web technology where information has a well-defined meaning (Berners-lee, Hendler, & Lassila, 2001). Another definition of SW is web content that has been described and associated in various ways to determine meaning and conditions by adhering to defined grammar and language structures (Hebeler, Fisher, Blace, & Perez-Lopez, 2009).

The concept of SW is not a coding language or program, but a layout. It is based on the preparation of data in certain formats to enable them to easily understand and interpret any data collection on their machines. In other words, the semantic web constructs the metadata of information in web content. SW technologies enable the classification of dense and irregular data existing in many fields such as marketing, distance education, health, finance to become understandable by the computer (Altay & Ulaş, 2018).

2.3.1 Semantic Web Standards

Today, most information and documents in the web environment contain a unique structure. Only specialized machines can understand and interpret this information set. This prevents communication between machines as a whole. In order to overcome all these problems and enable the machine to machine communication, World Wide Web Consortium (W3C) creates a cascaded structure in semantic network technology and presents a separate data infrastructure at each step, ensuring that the data at each step can be read by different machines (Övünç, 2004). Thus, machines can extract the same information from the data read in different forms and styles. Technologies such as XML, RDF, SPARQL, OWL are used in the creation of Semantic Webs. The layered architectural structure created by the W3C organization, chaired by Tim Berners Lee, in order to develop a standard in semantic network studies worldwide is presented in Figure 2.2.

Today, concrete studies on the layers of logic, evidence, and trust at the top have only begun to gain momentum. To summarize the duties of these layers that are related to each other;

(i) **In the logic layer,** some special rules are created for the relationships between ontologies (Berendt, Hotho, & Stumme, 2002).
(ii) **In the proof layer,** the rules created in the logic layer are proved to be correct.
(iii) **In the trust layer,** the results of the rules run in the evidence section are evaluated and the reliability of these results is discussed.

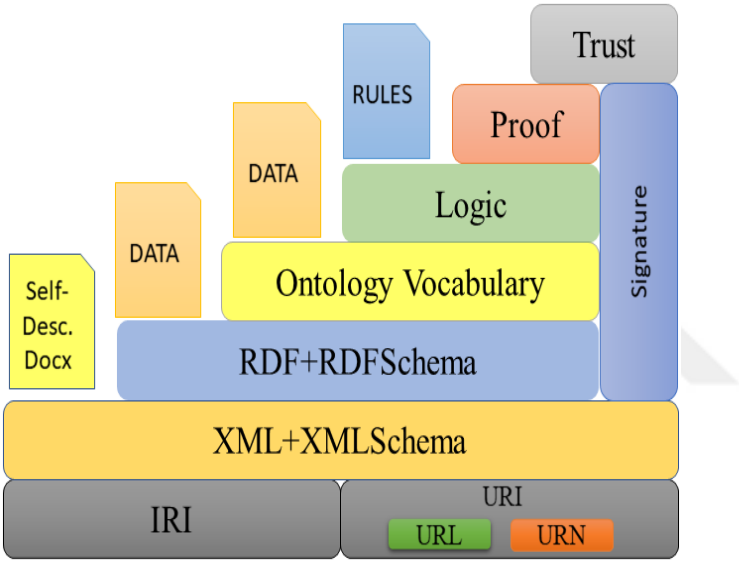


Figure 2.2 Semantic Web Layers (Miller, 2001)

If the rules established in the logic layer based on the relationships between the concepts are reliable, more consistent results can be obtained than in a classical search. However, when the rules in the logic layer are not reliable, it is possible to list results that are not relevant. The relationship in all semantic web layers is the same as in these 3 layers. In the creation of a useful and robust information frame of a domain, the consistency of each step within itself is only possible with the proper preparation and presentation of the data in the previous step.

2.3.1.1 Universal Resource Identifier (URI)

URI whose standards are determined by W3C is the character set that enables access to a resource (document, image, table, file, etc.) in the internet environment. It was first created and used by Tim Berners Lee in 1994 using the UNIX directory structure. URIs consist of the URL containing the address of the resource defined in another location and The Uniform Resource Name (URN) given to permanent objects in this location.

2.3.1.2 Extensible Markup Language (XML)

XML was developed by W3C because of, especially deficiencies in the common representation of data, data transportation, and accessing data, of Hyper Text Modeling Language (HTML) technology. The grand goal in the development of XML is to create web content that people and machines can understand. With this feature, besides storing data, it also serves as an intermediate format for data exchange between different systems (Balmin & Papakonstantinou, 2005).

2.3.1.3 Resource Description Framework (RDF)

RDF is a data markup language that enables data to be defined, structured, and presented. RDF is built on existing XML and URI technologies described in previous chapters. RDF is a family of W3C specifications and it is also used in knowledge management applications. (Punnoose, Crainiceanu, & Rapp, 2012).

A standard RDF file consists of resources, properties, and values components. It expresses an entity concept that is evaluated on resources. Properties are a type of attribute belonging to the resource, while values represent the value of the resource property. Resource and Properties are generally expected to be a URL, while Values can be in a URL or classic data type such as integer, string date. The example of RDF triple as the graph model is given in Figure 2.3.

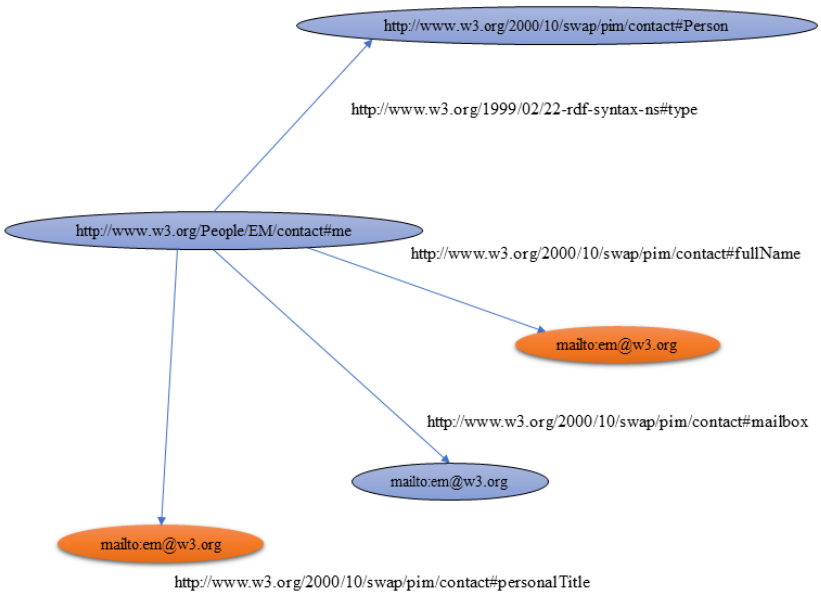


Figure 2.3 Graphical representation of personal information in RDF format

2.3.1.4 Ontologies

Ontology is the main subject area of philosophy that has been the subject of existence until the 1990s. However, in the 1990s, it has become a name that is frequently heard as the backbone of SW technologies in the informatics world. Ontology is a set of words that enables the information of a domain to be coded in a meaningful way, to share and to be used by machines.

Many definitions of ontology have been made and accepted since its emergence as an informatics term. In terms of the definition of ontology, the most accepted and used definitions in the literature are given below in chronological order.

- The ontology includes the basic terms and concepts that make up the vocabulary of a subject area and the rules for combining the terms and the relationships to be defined in order to expand the word string (Neches et al., 1991).
- According to Gruber, ontology is a clear representation of conceptualization. (Gruber, 1995).
- According to Borst, ontologies are common definitions of a shared conceptualization (Borst, 1997).

- Ontologies are a new generation technology that provides a common understanding of shared formal conceptualizations of certain domains, range, issues that can be communicated between people and application systems (Decker et al., 2000)
- Ontology can be used to describe and specify spatial data semantically and the machine in an understandable way (Wang & Kong, 2007).
- Ontology is the skeleton of knowledge about a particular field (Powel & Hopkins, 2015).

Since the RDF and RDFS languages are inadequate in some areas to describe ontologies, which are the heart of the semantic web, new high-level languages have been developed by improving the capabilities of these languages (Ekinci Eser, 2006). Ontologies are defined by specialized languages such as RDFS, DAML + OIL, OWL. These languages are used to define and diversify ontologies. With ontology languages, information is designed not only for users to understand but also for computers to process by making sense.

2.3.1.5 Simple Protocol and RDF Query Language (SPARQL)

RDF triples are the basic structure for accessing web content created in accordance with RDF standards. If users have an RDF resource, they can access other RDF resources using their features. Because RDF triplets establish relationships between resources. In this way, users can access the information they need as soon as possible. However, accessing information among a huge stack of data on the web wastes both time and labor. For this reason, many RDF or OWL query languages have been developed to examine or query web content with RDF standards. These are RDQL, Squish, Versa, SPARQL, etc. The query language used for querying RDF structures very similar to the SQL language. It is also a data access protocol for the semantic web. It helps to make inferences by making inquiries on ontologies.

SPARQL is used to extract the RDF graph from the endpoint. Just as SQL provides a standard query language in relational database systems, SPARQL provides a standardized query language for RDF charts or resources (Segaran, Evans, & Taylor,

2009). SELECT is the query type most commonly used by data consumers or developers in SPARQL query language. There are ASK, DESCRIBE, and CONSTRUCT query types besides the SELECT query type.

2.4 Data Mining Methods and Machine Learning Approaches

Today, advances in the collection and storage of information have led to an enormous growth of digital data. The continuation of electronic and digitalization efforts clearly shows that the rate of data increase in the digital environment will continue in the future. The fact that electronification efforts are being implemented almost every day in various fields has caused a great increase in the type of data available in digital media. This variety of data covers many areas such as personal data, health data, bank data, stock market data, sensor data, social network data, e-mail data, electronic marketing data, meteorology data, training course content, security data of companies and etc.

As the variety and volume of the stored data increased, it became difficult to infer meaningful conclusions from the data. In time, reaching the information that has the potential to be useful when it occurs in the data has become a problem that classical methods cannot overcome. As long as this data collected does not turn into meaningful information, it is worthless. Information contained in bulk in computer environments can be transformed into meaningful information by processing with data mining methods (Yılmaz, 2009).

Converting data into meaningful information is a process. This is the process of obtaining previously unknown but useful information and patterns from large volumes of electronic data. Data mining has created a new data analysis method by bringing together information technologies, statistics, machine learning, database technologies, and other related disciplines (Gemici, 2012). Data mining methods include various technical approaches such as summarizing data, clustering, analysis of changes, detection of deviations (Vahaplar & İnceoğlu, 2001).

For applications that have to use large data heaps, integrating data management methods into the application has become a necessity rather than optional. The subject area of data mining, which has been studied intensively since its emergence, has been defined similarly or differently by many researchers and scientists. In the literature widely accepted and many similar definitions of data mining are listed below.

- The discovery of previously unknown and understandable and useful patterns of particular importance from large data sets (Büchner, Anand, & Hughes, 2014)
- Data mining is to obtain information from the data set to be used and to reveal it to be used in an understandable structure (Ganesh, 2002).
- Data mining is the process of revealing hidden information in data stacks consisting of many situations and variables and transforming data into decision support-based information by using statistical analysis techniques and artificial intelligence techniques together (Yılmaz, 2009).
- The purpose of data mining is to create decision-making models for predicting future behavior based on the analysis of past activities (Koyuncugil & Özgülba, 2009).
- Data mining can also be defined as the finding of information in databases (Knowledge-Discovery in Databases) using interdisciplinary techniques of computer science (Bilgin & Acun, 2016).
- Data mining is to extract information from large volumes of data. In other words, data mining is the science of discovering unknown patterns, valuable structures, and interesting relationships between large and complex data in databases (Coenen, 2011).

2.4.1 Applications Areas of Data Mining

In today's business life, data mining applications are frequently used in finance, marketing management, education, engineering, industry, health, and many engineering fields (Ertugrul, Organ, & Savli, 2013). Data Mining (DM) can be easily applied in all areas where recording data is valuable for any system. The use of DM in areas that allow the creation of large data warehouses within the application makes a great contribution to getting more accurate results. More examples will make the

information extracted more meaningful. Data mining application areas are presented in Figure 2.4.

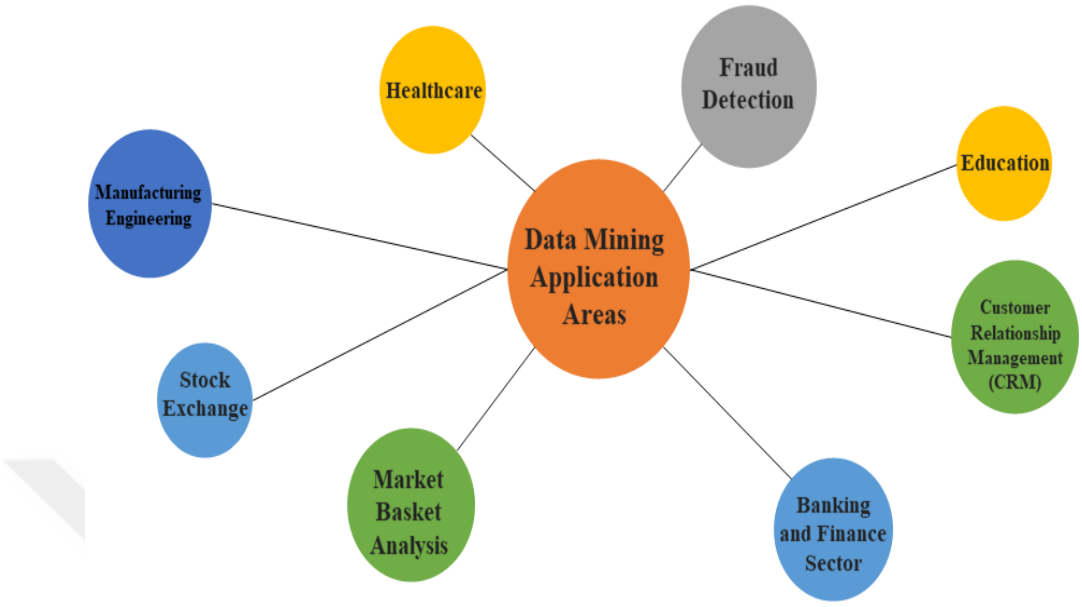


Figure 2.4 Some application areas where data mining approaches are used most

2.4.2 Data Mining Process

According to the definitions in the literature, DM is the process of obtaining meaningful information by using different methods of meaningless large data heaps. This process consists of steps independent of each other. Achieving successful results is directly related to the success of each step that constitutes this process.

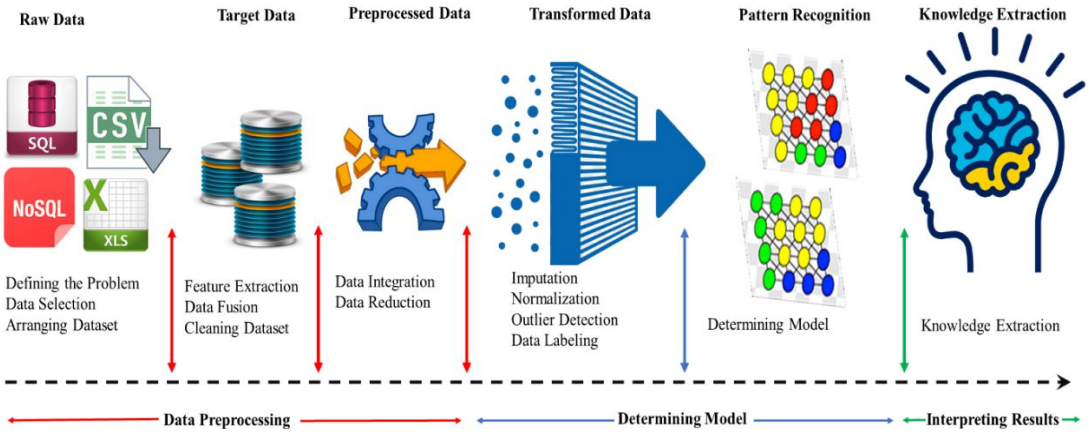


Figure 2.5 Presentation of Data Mining basic operation steps

Completion or failure of any of these steps is likely to negatively affect the outcome of the proposed DM process. These process steps are shown in Figure 2.5. DM application steps consist of mini processes, each of which has different methods. A proposed potential data modeling implementation process represents a combination of methods at these steps. There are numerous methods in the literature that can be applied at every step. For this reason, the data mining process to be applied for a dataset or case study requires serious work. These steps are explained in detail below.

2.4.2.1 Defining the Problem

The purpose of the work to be done is the stage in which its scope is determined. In other words, it can be called the initial starting point or planning stage of the project. The work to be done should be presented in general terms. The data set and parameters to be used must be determined. This proposed study is based on 8 different attributes including laboratory environment parameters. In this context, the goals of the proposed project is to establish a model that will ensure that measures are taken by predicting the analysis results and potential situations.

2.4.2.2 Collection of Data and Definition of Data Set

This stage covers the determination and collection of parameters suitable for the purpose of the proposed modeling. How long this collection process will take and at what intervals the data collection will be done should be evaluated within this scope. The data may be data collected by other researchers before, or they can be collected by the researchers who conducted the study. In the proposed study, 2 different environments for which measurements were planned were selected and these 8 parameters were measured with 5 different sensors.

2.4.2.3 Data Selection

This stage is the step of selecting the most suitable data for the purpose from the database. While this step is being executed, it includes removing any attributes that

seem unsuitable or worthless from their dataset. It is vital that unnecessary columns are not included in the data set, especially in studies that receive their data from other researchers. In this study, since the data collection process is a process planned within the scope of the project, the parameters and attributes required during modeling were collected and no action was required in this step.

2.4.2.4 Preparation of Data (Data Pre-processing)

This section is also called data preprocessing in many sources in the literature. The main purpose of data preprocessing is to increase the reliability of the data. Data preparation is one of the most critical stages before creating a model. Because how a healthy model is created depends on the data being included in the system in a ready and clean way. The data preparation process is a process consisting of several steps.

2.4.2.4.1 Formatting the Dataset (Arranging Dataset). In general, most of the work done in the field of data mining is the work done by using intermediary programs such as RapidMiner, WEKA or by writing their own codes by developers. In both ways, it should be ensured that the data given to the system is in certain formats. Because these programs support and can handle certain formats. Otherwise, healthy results cannot be produced or no results can be produced. For example, the WEKA program supports .arff or .csv formats. Therefore, in order for modeling with WEKA, it should be ensured that the data set to be studied is in these formats.

2.4.2.4.2 Cleaning the Data Set. Data obtained from different sources may not have the desired qualifications. It is important to delete or correct such data in order to get real results from the proposed model. Often confused with Outlier detection. In outlier detection, the data that may be in the data set but may surprise the model due to its very different value is eliminated, while In the Data cleaning stage, the data that cannot be in the data set due to the characteristics of the data set are cleaned. Data cleaning includes situations such as string type in any row in the data set due to some malfunctions in the system or the sensor measurement is outside the sensor range.

2.4.2.4.3 Data Integration. In data mining applications, a sufficient number of samples should be given as input to the system in order to guarantee healthy and consistent results from the model. However, sometimes similar data sets obtained from different sources can be combined because there are not enough samples in the data set to be worked on or in order to increase the accuracy of the system. As a result, although it will cause a slowdown in the performance of the system the more samples in the data set, the more likely it is to produce healthier results in the proposed model. In order to increase the accuracy of the model in this way, giving the maximum number of samples as possible despite an acceptable slowdown in the system performance provides many more beneficial results.

2.4.2.4.4 Data Reduction. Sometimes the data may be too much for the proposed model to handle. Depending on the parameters given in tree structures such as Decision Tree, Random Forest (Maximum Deep, Number of Trees, etc.), having a large number of samples for a large number of attributes may reduce the model's output to unbearable times. This is not acceptable especially for systems that have to work in Real-Time.

There are data reduction methods such as Data Cube, Dimension Reduction, Discrete, Sampling in the literature. If the data in the dataset consists of repetitive data in most places, one of the solutions is to reduce the data on a row basis to increase the performance of the system. In some sources, data reduction is called data summarization. The reason for this is that the average of data at certain time intervals (such as hourly, daily, and etc.) is entered into the data set as a single value. Although the minute data were collected in this proposed study, the hourly averages were added to the data set because the data were too repetitive.

2.4.2.5 Transformed Data

This stage is at least as important as the stage of preparing the data for the proposed model to reach the information that can be useful within the data stacks. Although the process of converting the data into formats that the proposed model can handle more

easily in the data preparation section, the data set is not ready for the full implementation of the model. In order for the data to be fully ready, it may be necessary to deal with the missing rows, eliminate some of the values that may affect the model too much, arrange the parameters in the data set the same range, and labeling the data according to certain criteria.

2.4.2.5.1 Imputations. Datasets often contain null data due to errors from human or machine source. This data is called a missing value in literature. The missing value may negatively affect the proposed model depending on its density in the data set. Therefore, it is important to fill the missing values with a logical approach, especially if approaches that are sensitive to missing values such as Decision Tree and Random Forest are to be studied.

In data mining, it is possible to solve the missing value problem with different approaches. One of the remedies is to fill in the missing values manually. However, if the Missing value is numerically excessive, filling it manually will cause time loss. Apart from that, deleting the missing values, accepting them as the average of that feature, or accepting zero is one of the most common missing value solutions. Many previous studies have shown that deleting or statistically filling missing values causes bias and negatively affects the result.

Therefore, imputing data can significantly improve the quality of the data set (Yang, Cheng, & Chan, 2017). Recently, many studies have shown that solving missing values with classification approaches has positive effects on the result (Abidin, Ismail, & Emran, 2018; Deb & Liew, 2016) . In the proposed study, approximately 90% of the data required to be collected over 45-days and 30-days periods were collected in two selected use-cases and recorded. The remaining approximately 10% could not be collected for reasons arising from human and devices (sensor, microprocessor, communication device, etc.).

2.4.2.5.2 Labeling. In classification and regression models, the values that make up the dataset are expected to have labels. This labeling process can be done manually as well as using some methods. Especially if the attributes to be used in modeling have distinct values that are accepted worldwide, it can be said that manual tagging instead of clustering methods makes the model results more consistent and meaningful. For example, in many studies around the world and according to the WHO, the hourly average of $PM_{2.5}$. value is 25 ppm. It is clear that this value, which is accepted in the literature, is more understandable and useful than a different value that can result from clustering.

In the proposed study, the reference values of the important parameters determining the laboratory air quality were determined by the WHO, EPA, ASHARE. In this study, these reference values were used while classifying the data and labeling them. Labeling is an important step to train the model correctly. Therefore, the labeling must be done as a result of comprehensive research. Sloppy or incorrectly labeled data can negatively affect the result produced by the model (Zhu et al., 2007).

2.4.2.5.3 Normalization. If the average of some parameters with their variants is too large or too small than the other parameters, and this great separation will have a greater effect on the others in the analysis steps and lower their roles to a significant value (Aydemir, 2017). The measuring range of each sensor used in this study is different. The measuring range is the total range that the instrument can measure under normal conditions. In the literature, there are normalization types such as Z Score Normalization, Min.-Max. Normalization, Ratio Transformation.

Absolute distance measuring methods such as Euclidean Distance, and Minkowski Distance, include them into the calculation with equal importance, if properties are in the same range. When using such distance criteria, calculating the similarity between instances without any pre-processing on the data set causes the feature with a large variance to have a high effect on the result (Jain, & Murty, 1999). In other words, a feature with a large variance dominates the effect of other features on the result. This is called Feature Domination.

Moreover, feature class labels with high variance may not have a positive correlation with the data at the point, that is, it may not have the power to distinguish data on the basis of classes. In this case, the classification process will be completely wrong. To avoid Feature Domination, (i) all features are shifted to a certain interval. (ii) Similarity criteria that are not affected by the feature domination problem such as cosine similarity can be used.

2.4.2.5.4 Outlier Detections. The outlier can be defined as any observation on the data set that is different from other observations in that data set (Barnett & Lewis, 1994). In the literature, outlier detection approaches such as Probabilistic, Distance Based (Cosine and Euclidean Distance, etc.), algorithmic-based (Neighbor, Neural Networks, etc.) are available. Outliers in the data collected by WSN can generally be caused by sensor measurement error or some problems arising from data communication. Sometimes outliers can arise from human error. Both system-based and human-based errors cause the estimation to be biased. Therefore, analyzing the collected data and eliminating some inconsistent parts will increase the power of the prediction.

2.4.2.6 Determining the Model

Data mining models are examined under two main headings. These models are Predictive Models and Descriptive Models (Akpınar, 2000; Zhong & Zhou, 1999). These modeling approaches are used for different purposes by using different methods and algorithms. There are also models referred to as "Semi-Descriptive", these models emerge as a result of using Predictive and Descriptive models together (Aydemir, 2018). In the modeling phase, it should be decided on what purpose an algorithm should be selected. Data mining models are given in Figure 2.6.

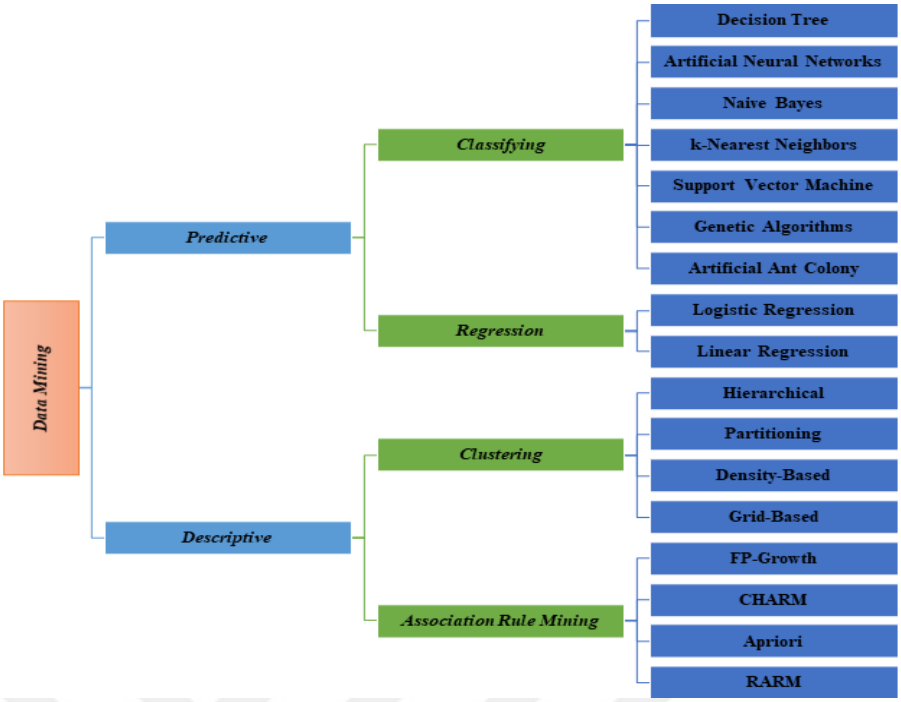


Figure 2.6 Commonly used data mining approaches

2.4.2.6.1 Predictive Model. In the predictive model approach, new models are developed based on the data whose results are known. Using this developed model, the results of the new data or the results of similar data sets whose results are unknown are estimated. For example, classification approaches are used to detect an anomaly. Transactions made by a bank customer through the internet and mobile banking channels are recorded. According to the customer's past behavior, a new model is created. The customer's new transaction of web and mobile banks is compared with the model created. A decision is made by estimating whether this new course of action belongs to the customer.

Classifying and regression models are the most widely used data mining techniques that are used in predicting the future based on existing data. The difference between these two methods is that the estimated dependent variable has a categorical or continuous value. Classifying and Regression methods are described below.

Classifying Methods: The process of placing data into a pre-determined appropriate group according to its common features is called classification. In order to be placed

in the appropriate class, the data must have at least one common feature determined with other data in the class. Classification, one of the basic areas of data mining, is defined in (Harrington, 2012) as a sub-branch of informatics used in placing an unknown data piece into a known group. Classification, which is one of the prediction models of data mining, and clustering, which is one of the descriptive models, is sometimes confused. The most important feature that separates classification from clustering is that class tags are given as input to the system before. In other words, in order to establish the classification model, it is necessary to know the predetermined situations and the values that the variables take in these cases. The class created by these values is called training data.

In classification problems, each element in the output space is called a class, and the algorithm that solves the classification problem is called a classifier (Camastra & Vinciarelli, 2015). Based on these definitions, a classification process is a process that enables us to reach the prediction class from the training class.

The first step in this process is the training set is determined and analyzed. Each element in the training set consists of different attributes that contain a label. The most suitable model is found according to the distribution of the data in the data set. This developed model is evaluated using the test set. In this evaluation, the label attribute is estimated by using other attributes that the apple has. There should be two data sets to create a classifier model. These are the training set where the label attribute is specified and the test set where the label attributes are estimated. With the training set, the algorithm is trained and the model is created, the model is validated with test data.

There are numerous classification methods in the literature. There are a variety of methods used for classification in data mining, such as NB Classifier, ANN, DT, SVM, k-NN algorithm. Apart from these, an artificial bee colony, which is suggested by inspired nature, is used in classification in heuristic algorithms such as genetic algorithms and ant colony.

Regression Methods: While classification algorithms are used to predict a categorical variable, regression analysis is used when estimating a continuous variable. For example, a classification model may be established for an insurance company to distinguish traffic insurance applications as safe and risky according to customer profiles. Regression models should be established in order to estimate the effect of a marketing firm's advertisements on television, newspaper, and radio on sales.

While classification algorithms are used to predict a categorical variable, regression analysis is used when estimating a continuous variable. For example, a classification model may be established for an insurance company to distinguish traffic insurance applications as safe and risky according to customer profiles. However, regression models should be established in order to estimate the continuous effect of a marketing firm's advertisements on television, newspaper, and radio on sales.

Regression Analysis is used to make predictions by applying formulas to existing data. The function is obtained from existing data using linear or logistic regression techniques. New data is used to make predictions by applying the existing function. In other words, this method is used to estimate other variables by using variables whose values are known. In regression terminology, the variable to be estimated is called the "dependent variable" and the variables used to predict the dependent variable are called the "independent variable" (Han, Pei, & Kamber, 2011).

2.4.2.6.2 Descriptive Models. In the Descriptive approach, another modeling method, they reveal the hidden relationships and patterns between the data that make up the dataset. In other words, the patterns within the existing data that can be evaluated in decision making are defined. The most well-known methods in this modeling approach are clustering and association approaches. Customer profiles formed by a newly established company by evaluating the parameters that affect consumer purchasing preferences and proposing potential products to the customer using these profiles can be given as examples of clustering methods.

Likewise, an e-commerce site identifying the products purchased together, determining the shopping habits of the customers and using it to design the site in a way to encourage the customer to buy more products can be given as an example of the association approach. Descriptive modeling approaches that are most used in the literature are Clustering and Association Rule Mining Methods. These methods are explained below.

Clustering Methods: Clustering is the process of dividing similar data into a heterogeneous data set into small groups. While the similarity of the elements between different clusters is less, the similarity between the same cluster elements should be high. In other words, a cluster is a collection of data sets that are similar to the data in the group it is in, but not similar to the data in the other group. The main purpose of clustering methods is to obtain homogeneous groups with similar features from a heterogeneous data set. Thus, it may be more efficient to work with homogeneously distributed small group data rather than working with a large heterogeneous group of data. In Figure 2.7, dividing a dataset consisting of heterogeneous data into more than one homogeneous group is represented as a representation.

When determining the set to which an element belongs, each record is compared with the existing sets and changes the descriptive value of the set to which it is assigned by assigning it to the set closest to it. The process is repeated until all records are optimally assigned to clusters. Therefore, clustering is a dynamic process. Unlike classification, the number of clusters that will result before clustering is uncertain. The number of clusters is determined by the characteristics of the data set. It does not require any prior knowledge about clusters before starting the clustering process.

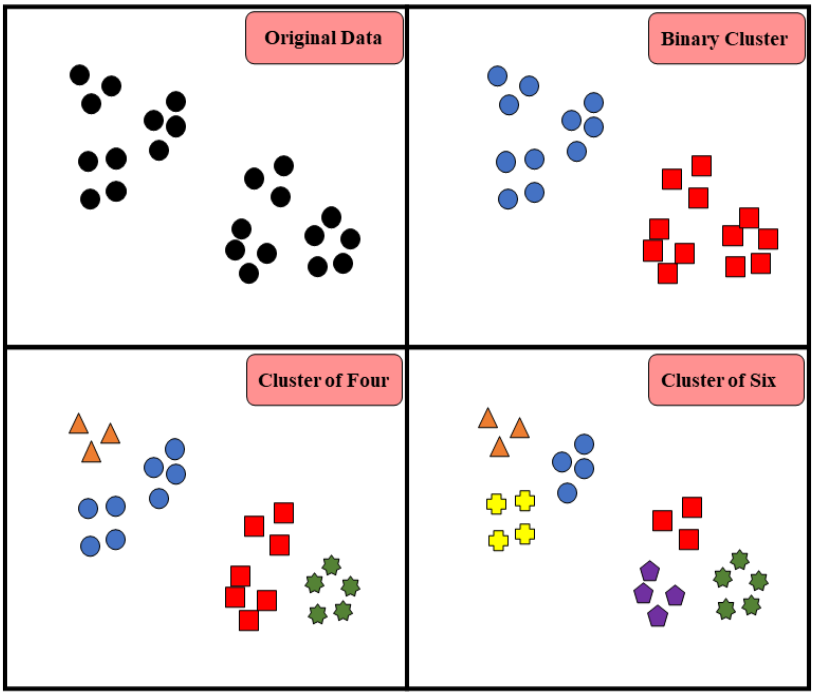


Figure 2.7 Division of heterogeneous data set into homogeneous groups

Clustering methods are used for many different purposes in various sectors such as statistics, chemistry, biology, sociology, machine learning, and marketing. In our daily life, clustering is used in many areas to make business processes more efficient. Classification of animals and plants, Classification of chemical elements and compounds, classification of vehicles, classification of houses according to certain characteristics are examples of clustering in daily life. Clustering methods are used for a data mining method in the informatics world, creating user profiles in social media sites, determining special marketing strategies in the e-commerce sector, and grouping documents on the internet.

Association Rule Mining Methods: Association rules mining are the method used to discover patterns that define relationships, in a large data set. The association rules method, which is one of the first methods that come to mind when it comes to data mining approaches, was first introduced and used by Agrawal, Imielinski, and Swami in 1993 (Agrawal, Almaden, & Swami, 1993).

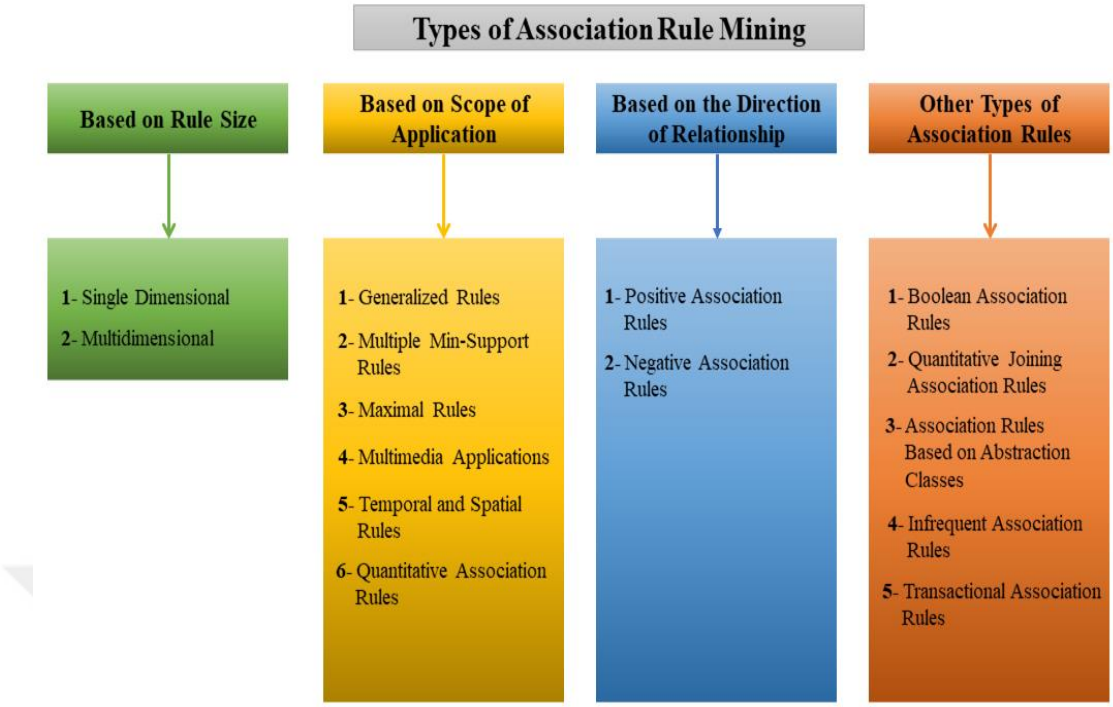


Figure 2.8 Association rules mining types

These methods reveal the rules of association with certain probabilities. Association rules are an approach that supports future studies by analyzing past data and determining association behavior within these data. It is possible to evaluate the types of association rules in 4 different categories according to the rule dimension, the scope of application, the direction of the relationship, and other association rules. Commonly used association rules types are given in Figure 2.8.

Although association analysis rules are used in many areas and applications, the most common usage area is market basket analysis. To give a concrete example of the association rules; It has been analyzed that 70% of the customers who buy bread and olives buy Mineral Water with these products. This analysis can be interpreted as that customers who buy these Bread and Olives have a high tendency to buy Mineral Water. Bread, Olive, and Mineral Water products must have been purchased together many times to detect such a union. Today, the fact that association methods are used so frequently in market basket analysis has caused the consumers to be known as Recommendation Systems, which offer products with higher purchasing potentials even though they do not need them.

CHAPTER THREE
LITERATURE REVIEW

Semantic web technologies were first used on sensor data by Avancha (2004). After this work by Avancha, the concepts of semantic unity and enrichment brought to the web world by the semantic network began to be used in sensor data. Since 2004, the concept of sensor ontology has become widespread day by day. Today, the concept of sensor ontology has become a key subject area for the understanding, interpretation, and reuse of sensor data, which has increased enormously with the diversification and increase of sensor-based automation systems. Moreover, semantic web technologies and ontologies suggest an appropriate approach to generate common words for sensor-based systems and to ensure the interoperability of sensor data from different platforms. With the work done so far, it guarantees that semantic web technologies and ontologies will be widely used on sensor data in the future.

Both the SWE and the SSN technologies for the definition of sensor data has the potential would be useful for possible disaster situations. Yang & Byun propose a semantic web-based framework to facilitate disaster management using distributed sensors (2020). In this project, to achieve higher efficiency in reasoning, the brain was inspired by the mechanisms behind synaptic plasticity. The proposed work focuses on the asynchronous spiking nature of sensors and extracts relevant temporal properties as seen in the processing of a neuron. In the other words, a scheme using spike-timing plasticity is proposed. As a result of their study, the main purpose of the developed STDP framework is to ensure the collection, sharing, access, use, and management of spatially organized data. Disaster response improves the judgment process by making all information available, accessible, and interoperable. The STDP system proposed in the study was tested and evaluated in a simulation scenario developed using MATLAB. The outputs of this project show that the proposed STDP framework can contribute to effective and efficient disaster management for time-constrained situations such as disasters, especially with regard to the timely implementation of action plans.

The fact that sensor networks gain great importance worldwide and are used in every application accordingly causes continuous observation data to be produced. Wang et al. (2018) presented a sensor ontology based on SSN ontology to describe sensor data obtained from heterogeneous hydrological network resources. In this study, the SSN framework is expanded by integrating time and space ontologies into the proposed sensor ontology. Time ontology of W3C is used for temporal concepts and the GeoSparql ontology of OGC is used for spatial concepts. As the last step of the extension of SSN ontology, the classes of specific terms belonging to the field of hydrology were created and these classes were concretized and ontological reasoning rules were determined. Their work was evaluated as a real-world use case in sensor data collected in the Yangtze lake, located in the southeast region of Wuhan, China. It has been verified that the study can recognize the various stages of flood events through semantic inquiry and knowledge acquisition experiments. Also, semantic queries are suggested hydrological sensor ontology can support the querying of heterogeneous sources.

In their article, Henson and et. al. (2009) were addressed two different issues for the representation of observational data. The first issue is that sensor data has a heterogeneous structure. The second issue is that there is no semantic proximity between sensor data. These problems may cause prevent interoperability and integration of the time series data collected from different sources. There are many different ways of representing sensor data nowadays. In other words, the representation of sensor data has a heterogeneous structure. Researchers evaluated their systems at real-world use cases in order to enhance meaningful their projects. Their proposed system was evaluated using sensor data monitored by the Australian CSIRO ICT Center. Overall, the researchers presented an ontological representation of time series observations in this article. They argued that SW would add a lot of value to time series sensor data on the Web (Henson, Neuhaus, & Sheth, 2009)

.

Goodwin & Russomanno (2006) proposed a prototype of the SSN system. Researchers have emphasized that integration of the raw data before storage of them, is necessary to extract more significant information than sensor data. Their ontology-

based prototype consists of wireless sensing capabilities, that include temperature, acceleration, GPS, light, barometer, magnetic field, acoustic measurements, and pressure (Goodwin & Russomanno, 2006). At the final of the paper, they argue that their prototypes involving the joint management of sensor information are successful and the concept of sensor ontology can be used for more complex systems. For future work, they are considering adding more sensors to the system, improving the accessibility and execution of sensor services.

Huang & Javed (2008) proposed a semantic-based architecture for the identification and processing of sensor information. In their paper, researchers emphasized that WSNs continuously collect massive amounts of raw data, which are generally only processed by customized applications. According to researchers, in order for applications and services to be developed independently of specific WSNs, the sensor data must be enriched with semantic web technologies and ontologies. Researchers carried out a potential fire accident for the use of a case study. They removed contextual awareness at the WSN and they provided to use of the sensor data easily with semantic web technologies for every application. This allows different consumers of the sensor data to provide more valuable services. According to them, standardization of semantic web technologies and sensor ontologies can help resolve this problem even further, in the future (Huang & Javed, 2008)

.

Janowicz & Compton (2010) present an overview of ongoing work to develop an ontology model of observation-based data obtained from the WSNs in their article. In the scope of this project, the core classes, relationships, properties, and another component of the model that forms the proposed ontology are discussed in detail. Stimulus, sensors, observations, observation properties, feature interests, procedures, and results of the system were explained by giving a variety of examples. Relations between these components were presented successfully in their study. According to researchers, the most important issue in sensor representation by semantic web technologies is that ontologies can be easily applied to any sensor-based domain. Finally, they point out that further studies will focus on documentation and the case to

show how to integrate ontologies or how extensions can be developed. In addition to other aspects, the relationship between sensors and results requires further study.

The rapid increase of data obtained from sensor-based systems brought the difficulty of managing data obtained from different systems in the same framework. Moreover, the lack of syntactic or semantic integrity between these sensor data made it difficult to share, reuse, and interpret them. Aktaş et al. (2020) have developed a standard data model for heterogeneous sensor data from different platforms by expanding SSN. The goal of this paper is to create laboratory environment parameters sensor ontology (LEPSO). In the proposed study, in 3 laboratories selected as measurement areas by the researchers, 8 different environment parameters are measured by, 5 different sensors.

A case study was conducted on laboratory environment parameters using real-time data collected from BAIBU SITARC. To evaluate the LEPSO sensor ontology, a series of semantic queries have been performed by the researchersThe results showed that sensor data, which is heterogeneous in nature, provides useful results in sensor-based tracking systems when enriched with semantic web technologies and ontologies. In addition, this study proves that the proposed semantic sensor ontology has the ability to provide a common infrastructure for many sensor-based applications. The proposed ontology has been claimed to have the potential to become a more comprehensive ontology by adding different platforms, different sensors, different environments such as schools and factories. In the next study, it is stated that this ontology is aimed to expand the scope of this SSN created by including a hospital's ICU.

Jin & Kim (2018) proposed an e-health system based on a semantic sensor network to solve interoperability problems of different platforms and devices. The system they recommend includes Expert user, Patient User, e-Health server, e-Health client, and e-Health device. They use the IETF YANG modeling scheme to represent information from the sensors they use for proposed e-health systems. This modeling scheme helps ensure semantic interoperability between devices and express detection data in a user-friendly way. According to the YANG modeling principle, the semantic model is

designed to include terminologies in the YANG modeling language. Ontology has been defined in YANG to create meta-models of E-Health sensors to provide a semantic interpretation of detection data in the system. It is argued in the proposed approach that e-Health sensors help to automatically configure and query the sensor network with semantic interoperability support for the e-Health system.

Semantic web technologies and ontologies propose a suitable approach for generating common words for sensor-based systems and to ensure the interoperability of sensor data from different platforms. However, these approaches are often not accepted by users and system developers based on sensors. This is due to the complexity of semantic techniques and the processing time to take longer than conventional methods. For the solution of these problems, Bermudez-Edo et al. (2017) have suggested IoT-Lite, a light example of SSN. Their ontology is an approach that provides interoperability of sensor data on heterogeneous IoT platforms and includes minimum concepts and relationships that can respond to most end-user questions in a reasonable time. To evaluate the proposed ontology, the researchers compared the performance of IoT-Lite with the IoT-A performance that another example of SSN ontology. In addition, in order to have more flexibility in ontologies, they brought the concept of dynamic ontology to the Semantic sensor network area. Dynamic Semantic sensor ontology usage example, MathML is used to store formulas and literal values. To demonstrate the usability of this dynamic approach, a case study was conducted using collective traffic data from Aarhus, Denmark. As a result of the case study, they proved that this approach provides faster response time.

Kuster et al. (2020) proposed sensor data model that would facilitate data transfer and eliminate heterogeneity between different sensor data. They suggested that this semantic data model supports urban sustainability close to real-time. The UDSA ontology has the ability to identify various sustainability key performance indicators, criteria, themes, and sub-themes in an urban system, as well as sensors and observations from perception. A case study was conducted in Wales to demonstrate the usability of this proposed sensor data model. A series of competency questions have been prepared to assess the reliability of the proposed ontology. In line with these

competency questions, they used the SPARQL query language, which allows them to query using classes, object properties, and data properties. In general, such a semantic model has proven to be effective in this study.

Onal et al. (2017) presented a weather clustering model enriched with semantic web technologies that were analyzed with machine learning methods and useful inferences were made. Moreover, in this study, the pattern recognition approaches of data mining and sensor anomaly detection were implemented in the system. In the evaluation of the proposed system, the number of clusters was limited in order to the easier interpretation of the results. It was stated that when more than 4 clusters were used in the evaluation phase, regions began to disappear, so the maximum k value was chosen as 4. The data analysis results show that it is possible to extract meaningful information from a relatively complex data set using the proposed system.

Adeleke et. al. (2017) introduced the ML-based estimation system in the SSW using stream reasoning in their article. Their model was evaluated in IAQ parameters monitoring in order to predict an unhealthy situation for the near future. In this project, the sliding window, which uses the Multilayer Perceptron (MLP) model to predict $PM_{2.5}$ pollution conditions, is integrated into their prediction model. The researchers tested the proposed ontology-based monitoring model in South Africa. The proposed system has been expected to help improve the IAQ, such as schools, and hospitals.

Three different home was selected for implementations of this test. Researchers placed sensor units in these three houses. During the test phase, the system tried to predict half-hour and one-hour future values of indoor air quality. The system decided the appropriate control actions, for implementations by occupants, if necessary. This control action was notified to the occupants by text message. Researchers have already identified the sliding window approach, which is also used in a number of time-series data studies previously, as a method of classification. Moreover, they have tried 5 different methods as classification algorithms. These are Bayesian Network (BN), Multilayer Perceptron (MLP), Decision Table (DT), J48, Random Forests (RF). The performance of the classification approaches was evaluated by the ROC method.

The data from the sensors were semantically enriched and ontology was created. This ontology was queried with the SPARQL query language integrated into the Eclipse development environment. The classifier was trained with the first 36 hours of data and a model was created. According to this model, the proposed system predicted the half-hour and an hour forecast horizon. The first 6 hours of data were removed from the training data and the new 6 hours of data were added to the training data. Every time the training data is changed, a new model is created and the predictions are updated. As a result of this work, Adeleke et. al. have successfully integrated Semantic Sensor Technologies' stream reasoning framework with machine learning algorithms in order to proactive monitoring and control. Moreover, in this study, researchers have proven that the short-term prediction of $PM_{2,5}$ can be successfully accomplished by an appropriately trained classifier.

The studies that were most similar to the proposed study in terms of technology and scope were evaluated in the third group. Studies under this group have also created a semantic-based framework for the definition of sensor information, and classical machine learning approaches have been performed on ontological sensor information. The biggest purpose of SSN is to create a common identification frame for sensor information from different platforms, different domains, and different sensors. However, in these studies, the number of platforms, sensors, and domains was limited and the capacity of SSN to represent sensor information in different systems, platforms, and domains could not be fully utilized.

In the proposed study, 2 different domains, 5 different environments, 4 different platforms, 5 different sensors were used and 8 different parameter values were measured. In previous studies, machine learning algorithms applied to ontological sensor data were limited in number, so in this study, the number of algorithms runs on sensor data was increased. Another difference is that many studies focused on either regression or binary classification. In this study, regression and binary classification approaches are evaluated together.

CHAPTER FOUR
MATERIALS METHODS

In this part of the proposed thesis, the technical hardware and software materials used throughout the study will be introduced. This section is of great importance as the materials, methods, and approaches preferred will directly affect the performance of the project. At the same time, cost-effective devices have been preferred in order to provide the sustainability of the system and to enable other researchers to perform similar studies. In other words, sustainable system design without sacrificing performance and accuracy has been the focus of the proposed thesis.

During the thesis work, microprocessors, communication devices, sensors, and various circuit components were used for the WSN design. In addition, Arduino IDE and Mysensors library are used for the embedded software of the nodes that make up the WSN. The controller that used to parse and manage data from sensors, was coded using a very commonly used software editor. The core structure of the proposed sensor ontology has been developed with the widely used Protege ontology editor. RDF database was used for storing the obtained sensor data and RDF query language was used for querying. For the analysis of the collected data, the RapidMiner data mining platform was used and appropriate prediction models were conducted for ontological sensor data. The remainder of this section includes the selection of materials and methods mentioned above.

4.1 Types of Hardware Equipment Used for WSN in Proposed Study

Indoor air quality (IAQ) measurement systems are generally integrated into local systems by researchers in areas where social health is important such as hospitals, schools, workplaces, and public transport vehicles. In this study, indoor air parameters data, which will endanger human health and affect the analysis results in critical areas such as the MCU and SITARC were collected with the help of WSN. The first step in setting up WSN is to design a hardware platform. The basic components to create a WSN are generally microprocessors, communication devices, and sensors. The

technologies used for hardware equipment and embedded system software used in the WSN created for the proposed system are described in detail below.

4.1.1 Microprocessor Board

With the integration of software and hardware technologies into industrial automation fields, the need for specially designed integrated electronic cards, which are generally used to manage these systems, has increased. Although it was difficult to program microcontroller cards in the recent past, with today's developing technology, cards that can be easily encoded by standard developers are now being developed. Among these microprocessor boards, the developers often prefer Raspberry Pi, BeagleBone Black, Msp 430, Freedom Development Boards.

However, in most of the studies carried out today, the most preferred card by the researchers is undoubtedly Arduino models. The main reasons why Arduino models are preferred so often are that they are cheap and easy to use compared to other boards. Arduino Uno is an ATmega328 (Microcontroller, n.d.) based microcontroller board manufactured by Atmel. Easy to use and affordable price is the reason to be preferred. The pin diagrams of the Arduino UNO and Atmega 328 used in the project are given in Figure 4.1.

Arduino is an open-source embedded system development platform that makes it easy to use hardware and software together (Baxter, Hastings, Law, & Glass, 2008). Arduino Uno was chosen as the microprocessor card in the node designs that make up the WSN, as it was deemed sufficient in terms of supply unit and analog-digital pin number within the scope of the proposed thesis.

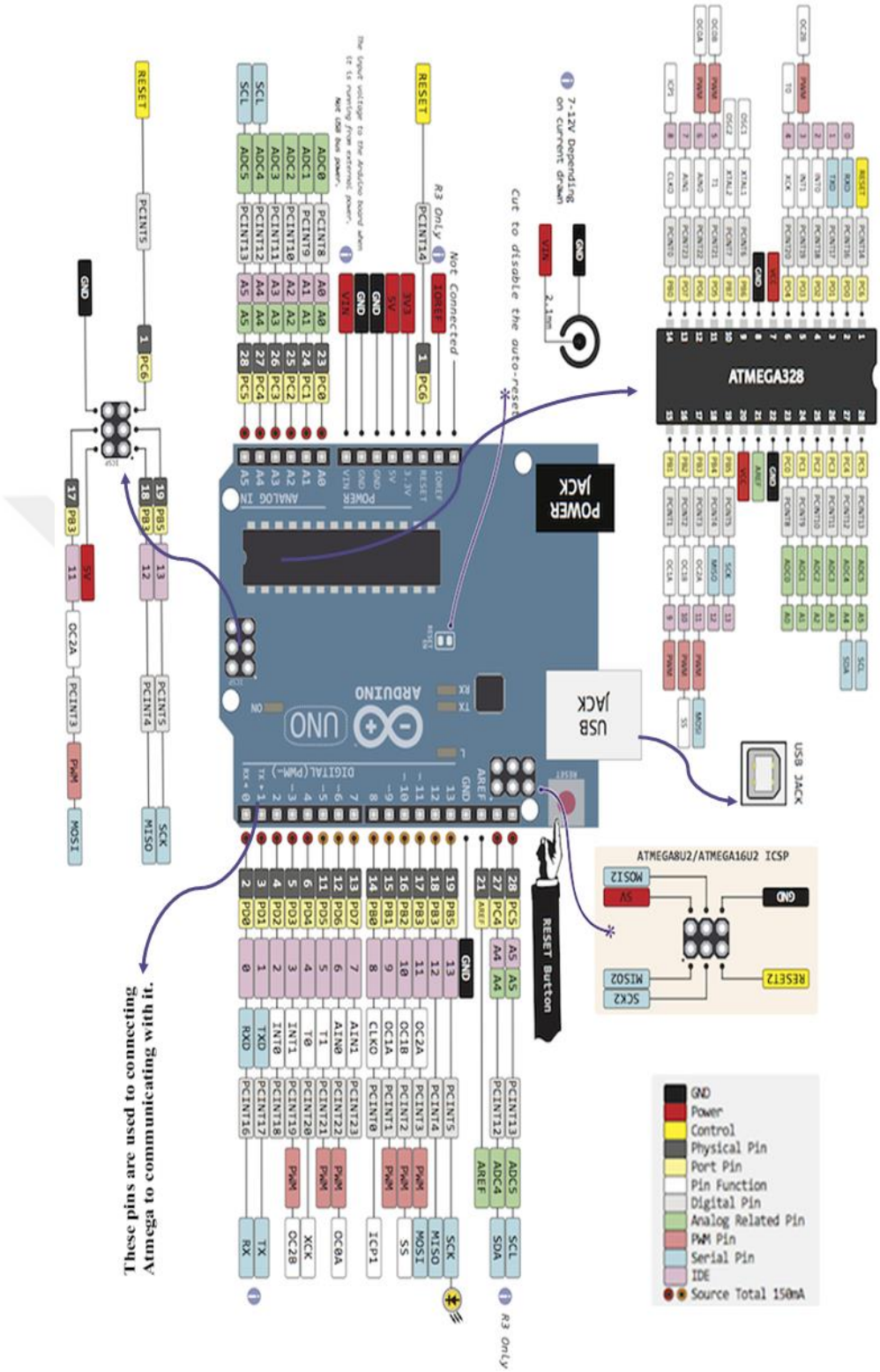


Figure 4.1 Pin diagram of Arduino Uno and Atmega328 (Pighix, 2013)

Table 4.1 Specification of Arduino Uno R3 development kit (Arduino, n.d.)

Specification	**Value and Comment**
Microcontroller	ATmega328P –8-bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5, range of 0V-5V)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz
Weight	25 g
Serial Pins	0(Rx), 1(Tx)

Arduino board is easily used in many areas from health, agriculture, security applications to robotic applications. Each application has its own characteristics. Different features and performances can be expected from the microcontroller card for each application. Arduino boards differ according to the need for supply, the number of digital-analog pins, the communication possibilities, and the physical size.

The most used Arduino models are; Arduino Uno, Nano, Pro Mini, Due, Mega, Leonardo, Lilypad, Esplora, etc. Besides, it is the decisive feature of the choice of this card in its compatible operation with many sensors, communication devices on the market. Arduino Uno R3 board is the most preferred Arduino board by both developers and beginners. The technical features and specifications of the Arduino Uno R3 card used as a microprocessor board in the proposed project are given in Table 4.1.

4.1.2 Communication Device

After the microprocessor is determined, the selection of a communication device compatible with the microcontrollers used to ensure smooth communication of nodes is one of the second and important steps of designing a sensor-based system. The most important feature of the device that should be selected as a Communication Device is its low battery consumption and its ability to provide seamless data transmission to a long distance. For these reasons, The nRF24L01 + PA/LNA Single Chip 2.4GHz Transceiver was selected and integrated into Arduino Uno R3 for WSN design in the proposed system (Nordic Semiconductor [NS], 2008).

Table 4.2 Specifications of the 2.4 GHz nRF24L01+ PA/LNA wireless module (NS, 2008)

Specification	**Value and Comment**
Frequency Range	2.4 GHz ISM Band
Maximum Air Data Rate	2 Mb/s
Modulation Format	GFSK
Max. Output Power	0 dBm
Operating Supply Voltage	1.9 V to 3.6 V
Max. Operating Current	13.5mA
Min. Current (Standby Mode)	26μA
Logic Inputs	5V Tolerant
Communication Range	800+ m (line of sight)

This communication device, developed by the Nordic Company, is a digital radio frequency wireless communication chip with low power consumption, which allows you to communicate wirelessly at the frequency of 2.4GHz, with both receiver and transmitter features. Technical specifications of the nRF24L01 wireless module are given in Table 4.2.

The nRF24L01 chip uses a built-in baseband protocol engine called "Enhanced ShockBurst" for ultra-low-power wireless applications (NS, 2008). nRF24L01 + communicates with microcontrollers via the SPI communication protocol. Since the

QFN20 is produced as SMD in a sheath structure, it is very difficult to buy and use this chip. Therefore, it is produced and sold as a module in the market. These modules are produced based on the reference module design found in the nRF24L01 datasheet. The pinout and scheme of the nRF24L01 + module is given in Figure 4.2.

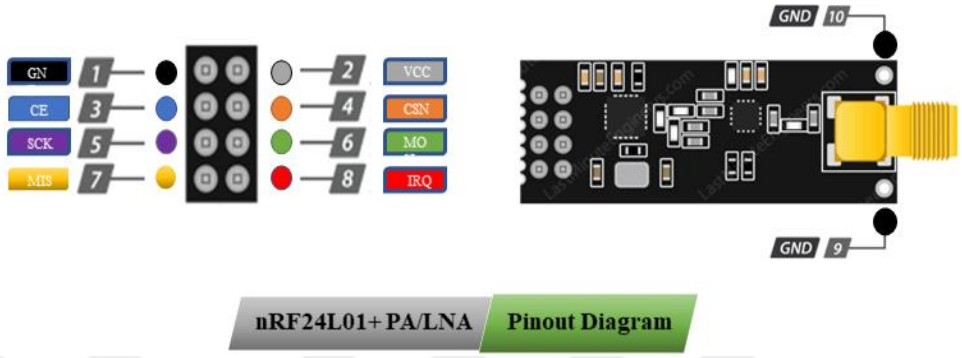


Figure 4.2 nRF24L01+ pinout diagram

The technical features and specifications of the nRF24L01 + module as a communication device in the proposed project are given in Table 4.2.

4.1.3 Sensors Used in Proposed Project

Sensors are devices that convert physical factors in the environment into electrical signals and become an indispensable part of robotic systems today. Sensors act as a bridge that connects the physical environment and industrial electrical / electronic devices. Within the scope of the proposed study, 8 parameters were measured by 5 different sensors and given as input to the system. These are DHT22 for temperature and humidity measurement, CCS81 to measure CO_2 and TVOC, Nova SDS011 to measure $PM._{2.5}$ and PM_{10}, MQ-7 to measure CO level, and LDR sensors to measure light level.

Proper sensor selection is an important issue to achieve project objectives. Otherwise, the measurements will not reflect the correctness, and the necessary action plans will not be effective. In the proposed project, an effort was made to select the most suitable sensor for the sensors selected for the measurement of environmental parameters, taking into account features such as error, precision, resolution, stability,

range, durability, response time, dimensions, cost. The features to be considered in sensor selection are briefly described below.

4.1.3.1 DHT22 Sensor Module

DHT22 is a capacitive and digital sensor that detects the temperature and humidity in the environment and can work stably for a long time. Small size and low consumption and long transmission distance (20m) make the DHT22 suitable for any demanding application situation (T. Liu & Manager, n.d.).

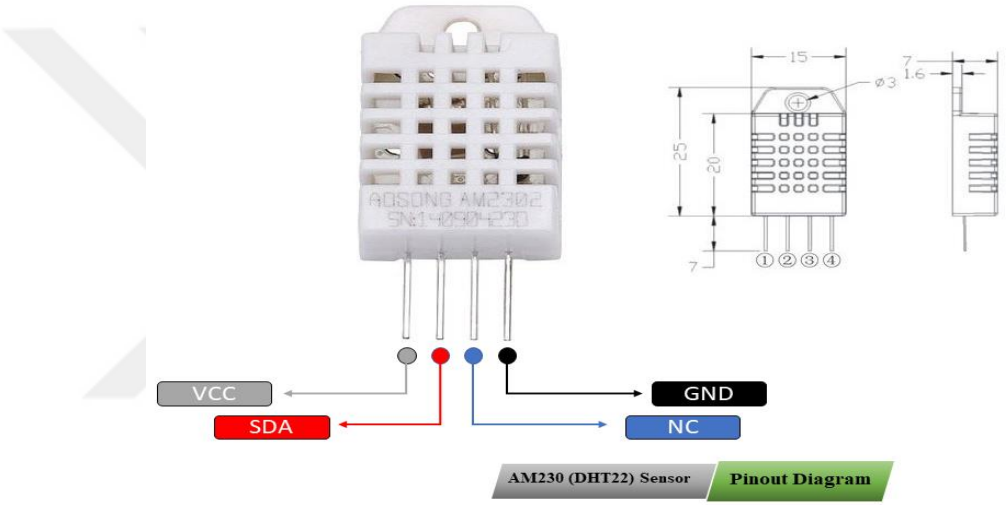


Figure 4.3 Pinout diagram of DHT22 humidity and temperature module

DHT22 sensor uses special digital signal acquisition techniques and moisture detection technology that guarantee its reliability and stability. It contains an 8-bit microprocessor and provides a fast and quality response. Average Response Time is less than 2 seconds. With its low power consumption and wide range, it is preferred by many researchers, especially in IAQ applications. The DHT22 sensor, whose pinout diagram is shown in Figure 4.3.

4.1.3.2 CCS811 Sensor Module

The CCS811 sensor module is a digital gas sensor with low power consumption that can detect the CO_2 and TVOC value in the environment. It is commonly used in

IAQ applications. With the help of a metal oxide (MOX) gas sensor integrated on it, it can detect a wide variety of VOCs in the environment (Ams, 2017). Thanks to the integrated microcontroller unit MCU, the sensor manages the driving modes and measurements itself while detecting all these VOCs. The pinout diagram of the CCS811 sensor is shown in Figure 4.4. It is capable of using optimized low power modes in restricted supply situations where there is no access to the grid.

Provides an indication of eCO_2 level or TVOC without host intervention. Includes analog to digital converter (ADC) and I2C interface. When connected to the microcontroller, it will return a Total Volatile Organic Compound (TVOC) reading and an equivalent carbon dioxide reading (eCO_2) over I2C (Ams, 2017).

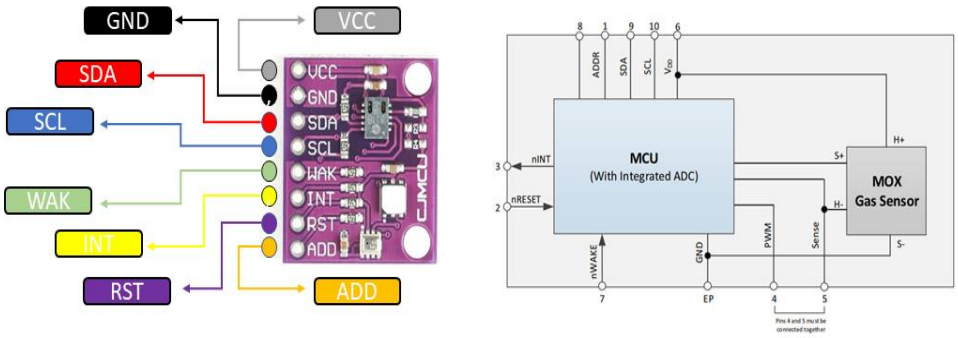


Figure 4.4 Pinout diagram of CCS811 digital CO_2 and TVOC module/sensor

4.1.3.3 Nova SDS011 Sensor Module

Using the laser scattering principle, Nova SDS011 can detect airborne particulate concentration from 0.3 µm to 10 µm diameters. This module is a highly accurate digital sensor with a built-in fan offering easy operation. This sensor module is used in many applications thanks to its reliable, stable, and consistent structure (Co, 2015). Another reason why it is preferred by the developers is that it can be easily integrated into every application thanks to its customized UART output. The pinout diagram of the Nova SDS01 sensor module is shown in Figure 4.5.

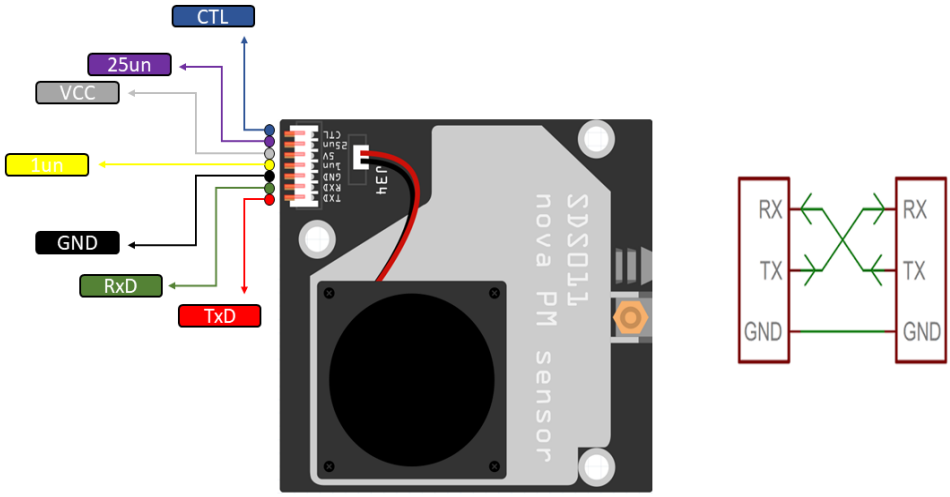


Figure 4.5 Pinout diagram of Nova SDS011 digital PM module/sensor

Unlike other sensors, one of the most important parameters showing the quality in a laser PM sensor is its service life. Because it usually contains laser diodes used to analyze dust particles. The quality of this laser diode determines the service life of the sensor. The life of the high-quality laser diode inside the Nova PM sensor is approximately 8,000 hours. It is recommended to use the default configuration when real-time data is needed (such as 1-second interval).

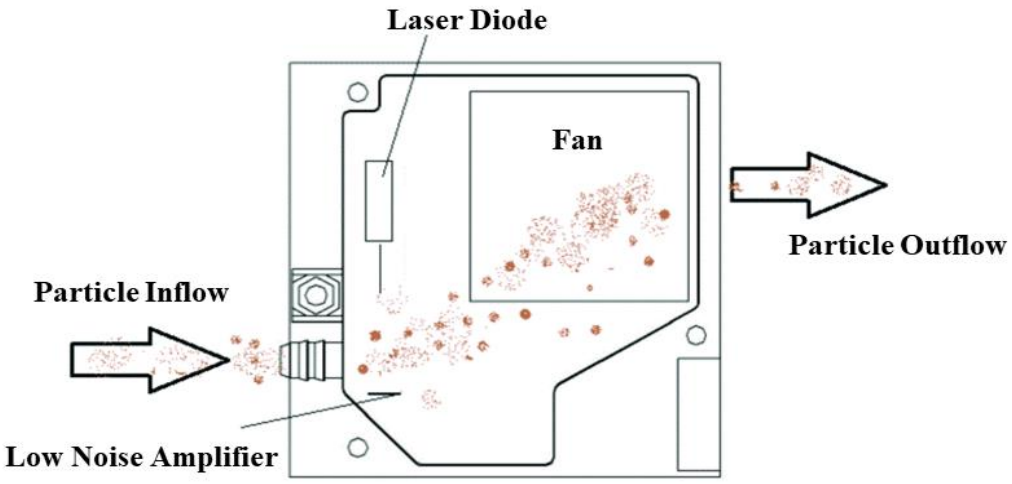


Figure 4.6 Working principle of Nova SDS011 digital PM module/sensor

However, if measurements are to be made at intervals of 1 minute or more, activating the sensor's sleep mode is certain to extend the life of the sensor. In addition,

activating the sleep mode when it is not needed will prolong the life of the network by preventing unnecessary energy consumption, especially in applications where there is no possibility to feed on the network. In this proposed thesis, measurements were made at 1-minute intervals. All sensors connected to the node were programmed to measure once a minute, allowing them to sleep for the remaining time. The operation principle of the Nova SDS011 sensor module is given in Figure 4.6.

4.1.3.4 MQ-7 Sensor Module

In the proposed project, it is planned to use the MQ-7 sensor module developed by Sparkfun for CO measurement. The MQ-7 is a long-life, low-cost carbon monoxide gas sensor that senses CO at concentrations of 10 ppm to 10,000 ppm and produces analog output (Electronics, 2018). Industrial CO detectors can be used as portable CO detectors for local gas leak detection. The MQ-7 Carbon Monoxide gas sensor can be easily used with micro control cards such as Raspberry Pi, Arduino, which are often preferred by developers for automation systems. The pinout diagram of the MQ-7 sensor module is shown in Figure 4.7.

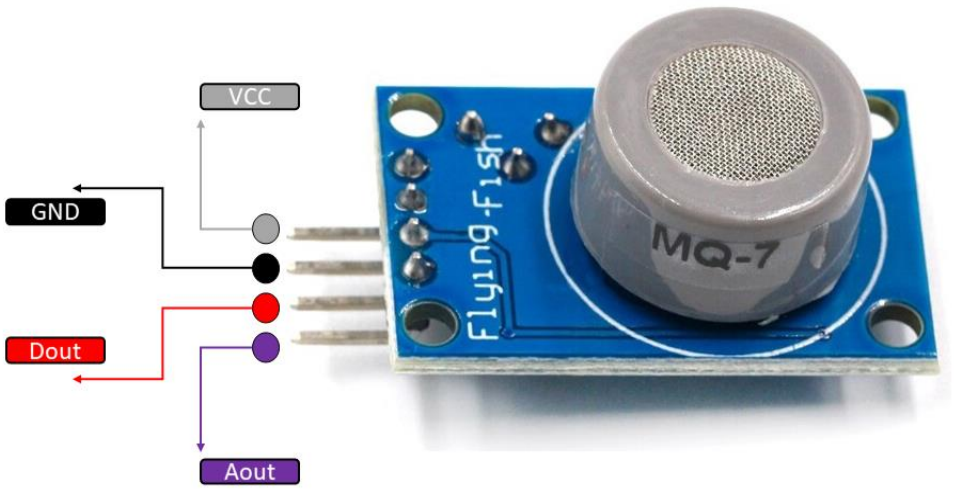


Figure 4.7 Pinout diagram of the MQ-7 analog CO module/sensor

4.1.3.1 Light Dependent Resistor (LDR) Sensor Module

Light sensors can consist of many parts due to their structures. Perhaps the most important of these parts is the light-perceiving component. The light-sensitive resistor

(LDR) is a widely used light sensing circuit element. LDR is produced from semiconductor and high resistance material due to its structure. Usually, cadmium sulfide (CdS) is used as a semiconductor. Two photoconductive cells with spectral responses similar to those of the human eye are used (Sunrom Technologies, 2008). The connection diagram of the LDR sensor used for light level measurement in the proposed project to the MCU is given in Figure 4.8.

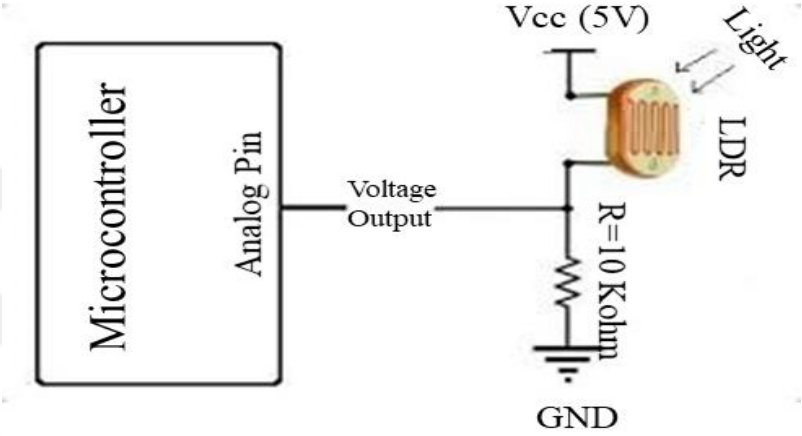


Figure 4.8 Connection diagram of the LDR sensor to MCU

LDRs provide an output with varying resistance values in the circuits where they are in. Since they perform this process with a physical change they receive from the external environment, they act as a passive sensor. Cell resistance increases as the light intensity decreases, or conversely, as the light intensity increases, the cell resistance decreases. The energetic photons (light) falling on it transfer their energy to the electrons in the valence electron band (high resistance), allowing them to jump into the conductive area.

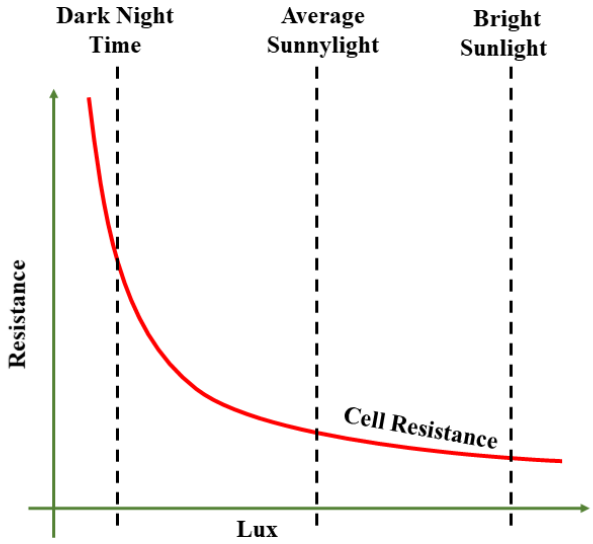


Figure 4.9 The relationship between the LDR internal resistance and the light

In this way, the resistance of the semiconductor material starts to decrease with the entanglement of more energetic electrons in the conductive region. As the decreasing light intensity increases, the number of electrons that jump to the conducting band increases, and the resistance of the material decreases. There is an inverse proportion between the internal resistance of the LDR and the amount of light falling on it. The graph of this relationship is given in Figure 4.9.

LDR, also known as a photoresistor, functions almost the same as the photodiode and phototransistors found in sensors. However, it is different from these in structure. LDR is in passive structure and creates resistance change as a result of light perception; Photodiodes and phototransistors also detect light with the help of PN junctions. LDRs can generally be used in applications such as smoke detection, automatic lighting control, product counting, and burglar alarm systems.

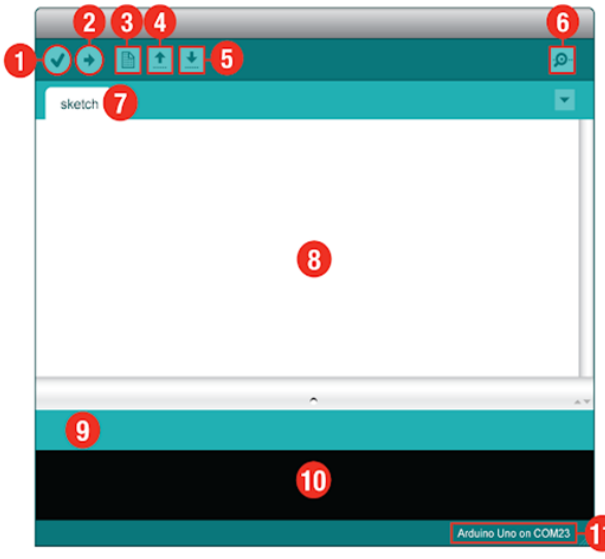
4.2 Embedded Systems and Controller Software Equipment

With the increase in digitalization and electronification activities worldwide, it has led to an increase in embedded systems designed to perform a specific process. The embedded system is microprocessor-based hardware with software designed to perform a specific function. Examples of embedded systems include ATMs, printers,

copiers, air conditioners, medical equipment, floppy disk drives, portable computers, game consoles, functional watches, and mobile phones. Within the scope of the proposed project, the Arduino IDE platform was used to provide embedded system software of sensor nodes. The MySensors library was used to ensure the communication and synchronization between the nodes and to integrate the sensors used into the embedded software.

4.2.1 Arduino IDE

In the proposed system, the coding of the node microprocessors is made with Arduino IDE. Arduino provides an open-source software (IDE) for programming the hardware. All Arduino boards and software are fully open source and allow for multi-platform support (Arduino, n.d.). In other words, different versions can be created by modifying the Arduino IDE. The Arduino IDE platform is offered and used without any limitations on functionality, operability, or usage. Arduino IDE is preferred in the proposed study due to its ease of use, open-source code, and too much documentation. The user interface of the Arduino IDE programming editor is given in Figure 4.10.



No	Button\Area\Display
1	Verify Button
2	Upload Button
3	New Document Button
4	Open Button
5	Save Button
6	Search Button
7	Sketch Tab Menu
8	Code Area
9	Debugging Area
10	Error Message Area
11	Arduino Type and Port Display

Figure 4.10 The user interface of the Arduino IDE programming editor

Programming with the Arduino IDE is done with a framework called Wiring. Based on the C ++ programming language in 2003, this framework is an open-source

programming framework for microcontrollers that started to be developed by Hernando Barragán, a graduate student in Italy (Barragán, 2004). Wiring allows you to write cross-platform software on a wide variety of microcontroller cards to create projects, device interactions, or developers' physical experiences in any field. The Wiring framework, which Hernando Barragán created so that designers and artists could approach electronics and programming more easily, led to the foundation of Arduino in 2005.

All developers who are familiar with the C ++ programming language can learn the Arduino IDE program more easily. When any program is written and installed on the hardware, the codes written in C ++ framework are compiled by Uploader named AVRDude. The codes are converted to the HEX file after if no problems with compilation. In the last stage, the program developed by communicating directly with the AVR-based microcontroller in a certain protocol and loading the HEX file through the USB interface is transferred to the hardware.

4.2.2 MySensors Library

There are many libraries written to establish the WSN network used by developers. Within the scope of the proposed project, the sensor nodes that will collect the data from the environment, the environment measurement software, and the embedded system software that will provide communication between them, MySensors library was used. MySensors is a free library for wireless IoT devices that allows devices to communicate using radio transmitters (MySensors Library, n.d.). The biggest reason for choosing the Mysensor library is that the documentation is easy to understand. Mysensors library is an open-source API developed by Alexander Budnik.

This library was originally developed for the Arduino platform only. Over time, the Mysensor Library has evolved into a framework designed especially for the establishment of WSN with microprocessors and communication devices such as ESP8266, Raspberry Pi, NRF24L01+, and RFM69, which are widely used by

developers. This library has the capacity to create a mesh and tree-like network structure supporting 254 nodes at the same time.

The MySensors devices create a virtual radio network of nodes that automatically a self-configuration. Each node can transmit messages to other nodes to cover greater distances using simple short-range transceivers (MySensors Library, n.d.). In other words, if the nodes in the network can reach the Gateway, they send the message directly to the Gateway. However, when they cannot reach the gateway due to distance and obstacles, a route that enables data to reach the gateway via other nodes that they can reach is followed. The Mysensors library provides the following facilities to developers.

- Providing create embedded system software of nodes in the network. (Serial Gateway, Repeater Nodes, Sensor Nodes)
- Ensuring give unique identification to every node in the network, Thus, the controller can easily understand from which node the data is sent.
- Enables give unique identification to every sensor on nodes. Thus, the controller can easily understand from which node the data is sent.
- Providing communication between whole nodes by self-configuration.
- Allows the nodes to find the shortest path to the gateway automatically.
- Providing the establishment of a dynamic network using repeater nodes or other sensor nodes even if the position of the node changes.
- Saves power by allowing sensor nodes apart from repeater nodes to operate in sleep mode. Thus, it extends the life of the sensor nodes and network.
- Supports 254 nodes in a single network and every node maybe include 254 different sensors. This is theoretically mean that 64516 sensors are managed in only one network. This number will be enough for many applications.
- In very large applications such as pipeline monitoring or structure monitoring, where the number of sensors or nodes is not sufficient, a parallel radio network can be created from the 124 useable channels.

4.2.3 Microsoft Visual Studio

In the proposed project, the controller interface has been devised for parsing, managing operations of messages from WSN, and real-time recording of sensor data into RDF databases. The controller interface is designed using the Visual Studio editor. Visual studio code development editor is used to develop console and graphical user interface applications along with Windows Forms applications, websites, web applications, and web services. (Chowdhury, 2017). The integrated debugger can perform both source-level and machine-level inspection. This enables the code editor and debugger to support almost all programming languages. With Visual Studio, applications can be developed infrequently used programming languages such as C #, VB.NET, C / C ++, F #. In the proposed project, the user interface was developed in C # programming language.

4.2.4 dotNetRDF Library

DotNetRDF is used for the definition of the proposed sensor ontology. DotNetRDF is an open-source RDF API used in Microsoft Visual Studio for implementing Semantic Web concept (Mishra & Singh, 2016). DotNetRDF library was written in C# designed to provide a simple but powerful API for working with RDF data (Barbur, Blaga, & Groza, 2011). This Library provides a large variety of classes for performing all the common tasks from reading & writing RDF data to query over it. The Library is designed to be highly extensible and supports for users to add in additional features (e.g., custom RDF Triple Stores) as required (DotNetRDF, 2020). The core classes of the Library can be found in the VDS.RDF namespace. All the core classes are based either on interfaces or abstract classes to make the library as extensible as possible (DotNetRDF, 2020).

4.3 Standardization Studies for Raw Sensor Data

Sensor data has been collected and represented in different formats and methods in many studies. In literature, most of the researchers collect and represent sensor data as time series observation (Bhandari, Bergmann, Jurdak, & Kusy, 2017; Sharmin et al.,

2015). Time-series observations are one of the widely used methods of collecting sensor data. A time-series data is the measurement of the feature of interest in succession over time. Over time, scientists have recognized the importance of quality time-series observations to conduct research and analyze data. For this, the data should be structural and cooperate. These similarities between sensor data and time-series data include sensor characteristics, measurement methods, data formats, measurement units, application area, the evolution of data in time, spatial solutions, etc.

Until today, sensor data may be published as only values on the internet, but searching, filtering, reuse, integrating, interpreting, and sharing efficiently these data requires more than just the indicate observation results. Sensor data is heterogeneous in nature because it is used in different systems with different syntaxes, structures, and meanings (Baxter et al., 2008). Moreover, the integration of the sensor data can be very challenging, especially when heterogeneous data sources are available in the WSN used. For this reason, processing and managing sensor data are getting more difficult day by day due to the lack of a specific standard for heterogeneous sensor data.

Until today, sensor data may be published as only values on the internet, but searching, filtering, reuse, integrating, interpreting, and sharing efficiently these data requires more than just the indicate observation results. Sensor data is heterogeneous in nature because it is used in different systems with different syntaxes, structures, and meanings (Sheth, 1999). Moreover, the integration of the sensor data can be very challenging, especially when heterogeneous data sources are available in the WSN used. For this reason, processing and managing sensor data are getting more difficult day by day due to the lack of a specific standard for heterogeneous sensor data.

The heterogeneity of sensor data causes these data to remain application-specific and different sensor-based systems not to be managed under a common infrastructure. An intermediate layer that will enable the sensor data to be enriched semantically and made more useful regardless of the application is a vital need. Recently, researchers claim that semantic perceptron web technologies can enrich raw data obtained from

sensors in terms of semantics and fill this middle layer (Haller et al., 2018; J. Liu, Li, Tian, Sangaiah, & Wang, 2019; F. Wang, Hu, Zhou, Hu, & Zhao, 2017).

In addition, a common framework is required for sensor-based information systems. Sensor data must be defined using URIs and sensor data must be transmitted to consumers over HTTP (Patni, Henson, & Sheth, 2010). Besides, sensor data must be encoded in machine-readable formats such as RDF and OWL so that it can be easily read and processed by machines. However, at this point, the lack of a comprehensive and comprehensible standard for the enrichment of sensor data worldwide has been the biggest problem in the common manageability and operability of sensor systems.

Sensor data must be standardized so that they can be successfully interpreted, application-independent, and reused in a variety of applications. Some standards had been developed with the aim of closing this deficiency in the literature. Two different standards are mostly used in many studies by researchers for sensor data. These are (I) SWE and Observations and Measurements Language (O & M) that was developed by the OGC and (ii) Semantic Sensor Network which was developed by W3C. In the following chapter, the architecture of both technologies is given briefly and discussed which of them more beneficial for this thesis study.

4.3.1 Sensor Web Enablement

The first standardization initiative is SWE and O & M which was developed by the OGC. SWE, which is members of the OGC, architecture model was developed as a common framework for the be able to implementation of interoperable and scalable service-oriented working of heterogeneous sensors data. In web-based sensor networks sensor location is usually a critical parameter for sensors end-user or system analyzers, and OGC is the world's leading geospatial industry standards organization. Moreover, OGC has enabled sensor systems to serve over the web.

The general purpose of SWE is to provide that any sensor, actuator, device, and camera accessible from the Internet is accessible and, where applicable, controlled

over the Web. SWE common language or standard is an XML-based model for representing raw data obtained from sensor nodes on the Web (Henson et al., 2009).

4.3.2 Semantic Sensor Networks

The second standardization initiative is the Semantic SSN which was developed by W3C. Established the SSN-XG in 2011 to fill the interlayer that will provide a common representation of the consortium sensor data and set a set of standards for the sensor. SSN-XG has done many studies and determined certain standards for semantic enrichment of raw sensor data obtained from sensor-based systems. The latest version of the SSN, which is still used as a common framework in many studies today, was published in 2017 (Lefort, et al. 2017).

W3C is a consortium that determines the web and semantic web standards in the world. The SSN that is put forward by W3, ontology can ability integration and high-level descriptions of sensor observation data. Moreover, this ontology can describe the capabilities of sensors that were set up, the measurement processes used, and their observations.

SSN has a core ontology called SOSA (Sensor, Observation, Sample, and Actuator) that including lightweight but self -contained, for its fundamental classes and properties (Compton et al., 2011). SOSA complies with the minimum interoperability limits, i.e., the sensor ontologies created with SOSA guarantees its sharing and interoperability with all SOSA/SSN ontologies (World Wide Web Consortium [W3C], 2017). SOSA/SSN framework modules are shown in Figure 4.11. SSN is a framework for providing meaning for sensor observations to provide status awareness. SSN enhances the meaning by adding semantic annotations to existing standard sensor languages. These additional statements provide more meaningful explanations and more access to sensor data than SWE and act as a link mechanism to SWE's gap between syntactic XML-based metadata standards and Semantic Web's RDF/OWL-based metadata standards (Compton et al., 2011).

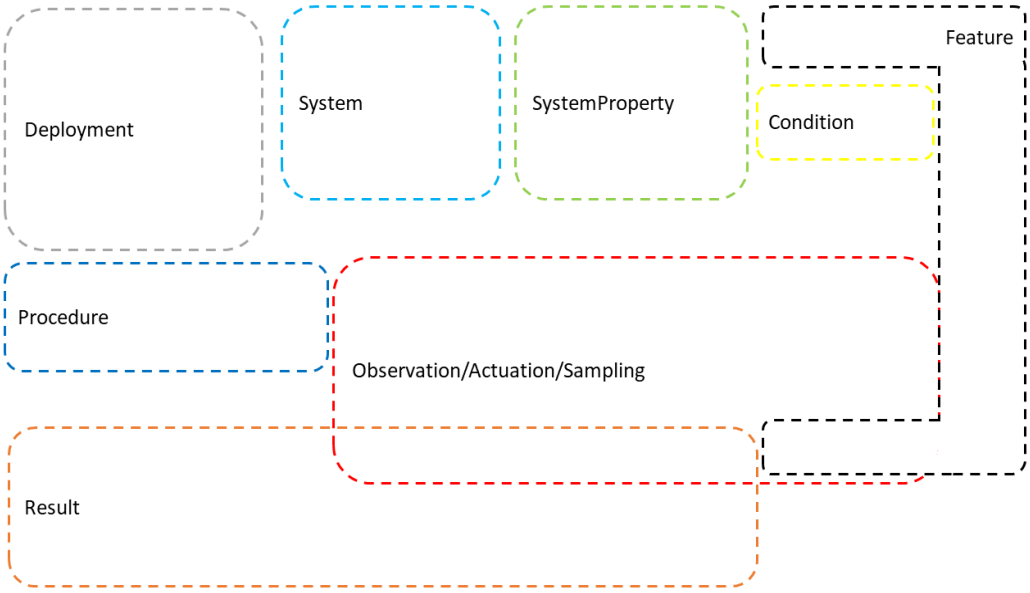


Figure 4.11 Overview of SOSA/SSN framework modules

Along with semantic descriptions, ontologies and rules play an important role in interoperability, analysis, and reasoning compared to heterogeneous multimode sensor data in SSW (W3C, 2017). The basic classes of SOSA ontology, which constitute the core of SSN ontology, the relationships between them are presented in Figure 4.12.

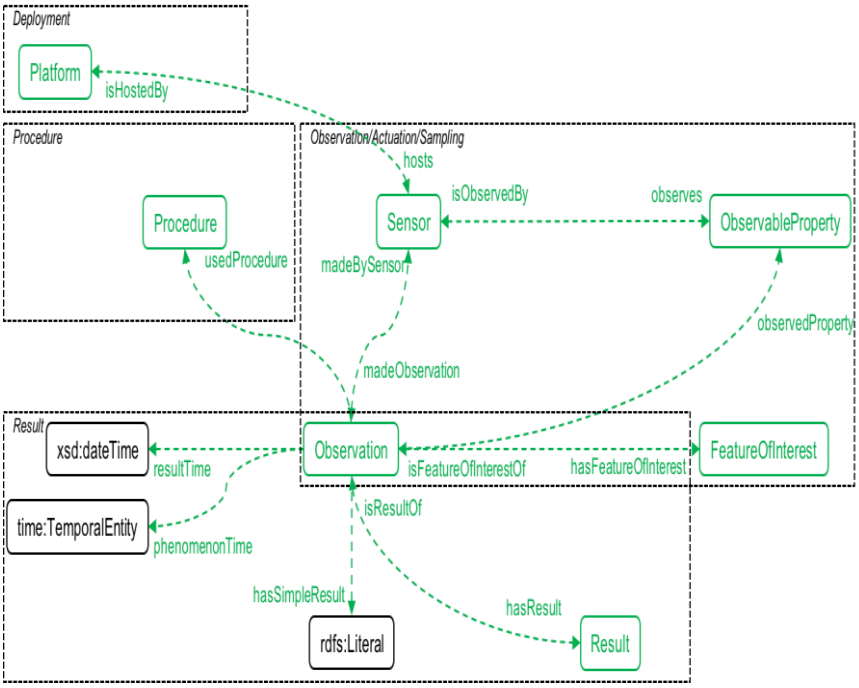


Figure 4.12 Overview of the SOSA classes and properties (W3C, 2017)

The semantic sensor network is an application-independent framework that must be expanded with specific concepts and examples (Calbimonte, Jeung, Corcho, & Aberer, 2012). SOSA/SSN frameworks are designed to allow its scope to be extended with other ontologies and concepts. For example, in an area ontology where geolocation is important, the ontology of terms representing location information can be integrated to extend the SSN core ontology. In another example, a domain ontology containing chemistry-related sensor measurements might import chemistry ontology including chemical terminology, classes, and object properties as an example of the expansion of the SSN core ontology.

Over time, firstly fundamental concepts, terms, and relations were developed like sensors, properties, observation, and systems in SSN and SOSA ontologies. After then measuring capabilities, operating and survival restrictions and deployments were added in these ontologies (Compton et al., 2011). Finally, SOSA/SSN frameworks have been aligned to the DOLCE UltraLite upper ontology (DUL) to interoperate use with their ontologies from developers using DUL aligned ontologies. The classes and concepts of the SOSA/SSN framework last version published by W3C in 2017 are shown in Figure 4.13. In this figure, the main components, classes, restrictions, and properties of SSN and SOSA Ontologies are illustrated together. In the figures, components of SSN ontology only are shown in blue, while related components of SOSA are shown in green.

Generally, within the framework of SOSA/SSN; deployment, system, platform, procedure, etc. conceptual modules that will form the infrastructure of sensor-based systems are defined. In addition, SOSA/SSN standards are a framework for defining sensors, actuators, sensor measurement capabilities, sensing observations, related procedures, observed properties, features of interest, and deployments. The most important and most used concepts, classes, properties of the SOSA/SSN framework, and the relationships between them are explained in detail below.

The classes and concepts of the SOSA/SSN framework last version published by W3C in 2017 are shown in Figure 4.13. In this figure, the main components, classes,

restrictions, and properties of SSN and SOSA Ontologies are illustrated together. In the figures, components of SSN ontology only are shown in blue, while related components of SOSA are shown in green.

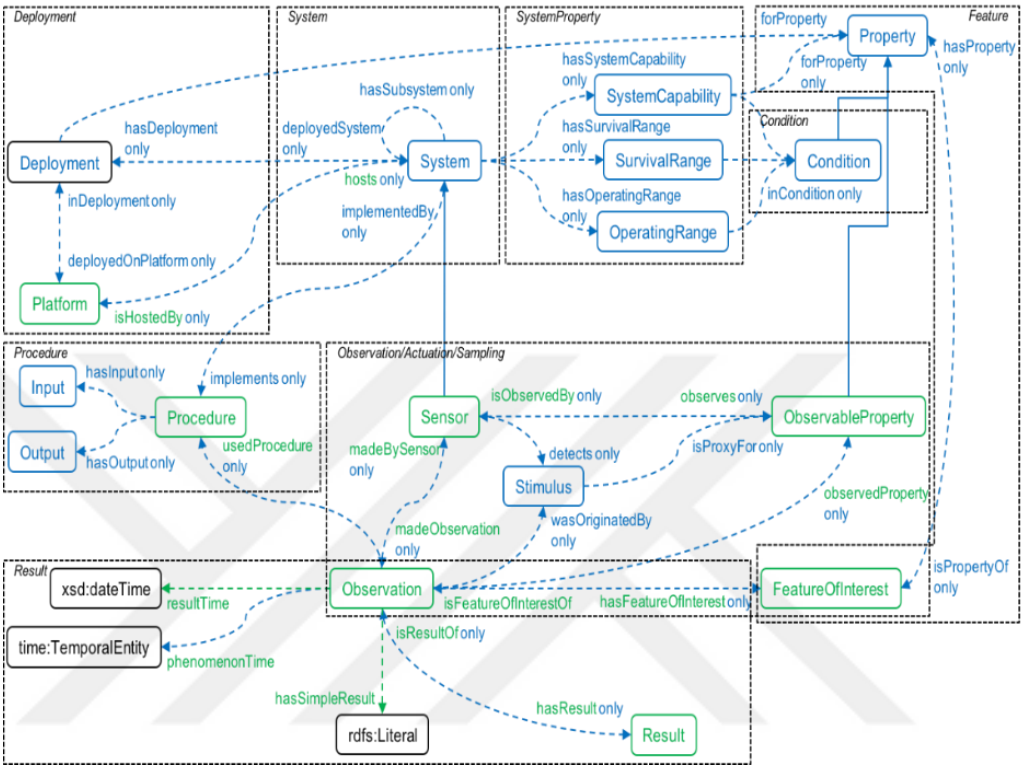


Figure 4.13 Overview of the SSN and SOSA classes and properties (W3C, 2017)

Generally, within the framework of SOSA/SSN; deployment, system, platform, procedure, etc. conceptual modules that will form the infrastructure of sensor-based systems are defined. Also, SOSA/SSN standards are a framework for defining sensors, actuators, sensor measurement capabilities, sensing observations, related procedures, observed properties, features of interest, and deployments. The most important and most used concepts, classes, properties of the SOSA/SSN framework, and the relationships between them are explained in detail below.

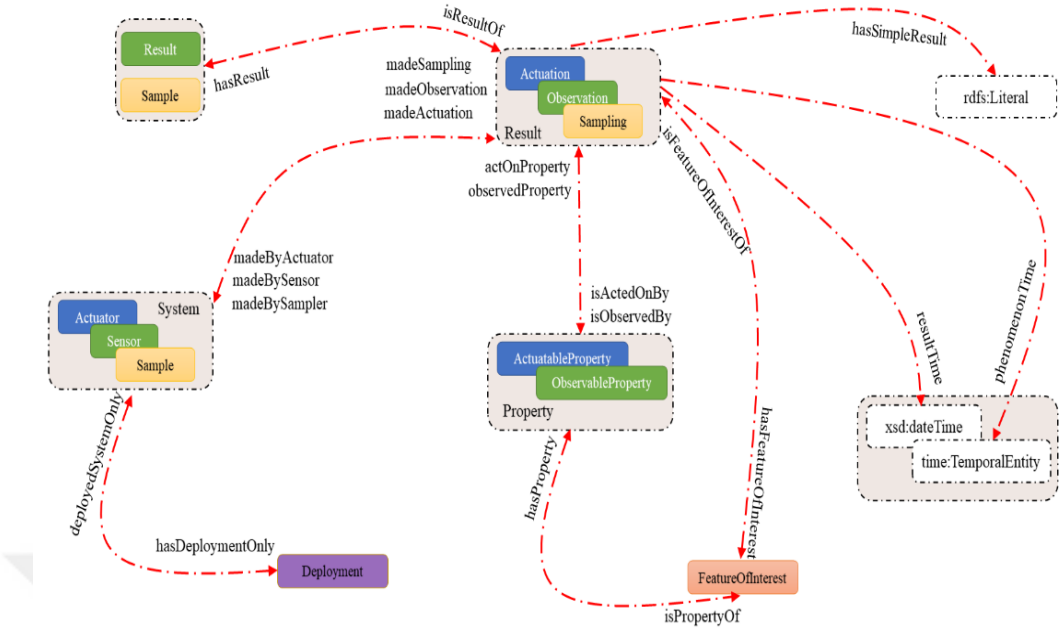


Figure 4.14 An overview of the SOSA/SSN from "ssn:Property" perspective

ssn:Property: The type of feature that is interested in. In other words, the aspect of a being to make it a being and cannot exist without this property. This Class divide into two subclasses named "sosa:ObservableProperty" and "sosa:ActuatableProperty". Each parameter measured within the scope of the proposed thesis study is a "sosa:ObservableProperty". For this reason, there are 8 "sosa:ObservableProperty". An alert subclass has been created under the "sosa: ActuatableProperty" class to inform the relevant personnel about the environment status. An overview of the SOSA/SSN classes and properties (Closely related to the property class) from "ssn:Property" perspective is given in Figure 4.14.

"sosa:ObservableProperty" is an observable characteristic or property of the "FeatureOfInterest" class. "sosa:ObservableProperty" is a value that can be directly measured or observed, such as The humidity and temperature value of an indoor environment the height of an object, or the concentration level of gas in the environment. Conversely, the value of a house or car is not a property that can be directly observed or measured. Their values are only asserted but cannot be measured directly. "sosa:ActuatableProperty" is an actuatable characteristic or property of the "FeatureOfInterest" class. A component in the indoor air quality system changing the

state of the ventilation system when the ambient conditions are insufficient may be given as an example of "sosa: ActuatableProperty". The ability of the ventilation system to be opened and closed is its "sosa:ActuatableProperty".

sosa:FeatureOfInterest: For any purpose, an object, feature, or environment in which, its "sosa:ObservableProperty" is measured by means of a sensor or whose "sosa:ActuatableProperty" is altered by an actuator. For example, when measuring the temperature of a room, the temperature level is a "sosa:ObservableProperty". 35 ˚C is the value or result of the "sosa:ObservableProperty". The automatic ventilation, and air conditioning control system in the measured room are a "sosa:FeatureOfInterest" for the Actuator. Within the scope of the proposed thesis, 2 "Sosa:FeatureOfInterest" was selected.

ssn:Deployment: Describes where "ssn:System" classes are located in "sosa:FeatureOfInterest" to measure any "sosa:ObservableProperty" or manipulate situation of any "sosa:ActuatableProperty". In other words, The place where the sensor or platforms are placed is called "ssn:Deployment" in SOSA/SSN ontology.

For example, to detect the CO_2 level in the environment, the wall where the CCS811 sensor is placed at the respiratory level of the people can be given as an example of "ssn:Deployment". A sufficient number of sensor nodes were placed in MICU, and LaEn that were selected as the measurement area. An overview of the SOSA/SSN classes and properties from "ssn:Deployment", “sosa:FeatureOfInterest”, and “sosa:Platform” perspective (Closely related to these class) is given in Figure 4.15.

sosa:Platform: Sometimes more than one sensor and actuator can be deployed in the same location. Then, members of the "ssn:System" class can be gathered on the same device to make the system easier to manage and to increase the efficiency of the monitoring process. These devices, in which more than one sensor and actuator are collected, are called platforms in SOSA/SSN ontology. A vehicle, mobile phone, computer, human or animal body, can be a platform for individuals of the "ssn:System" class, or it can be in a prototype created by the developer himself.

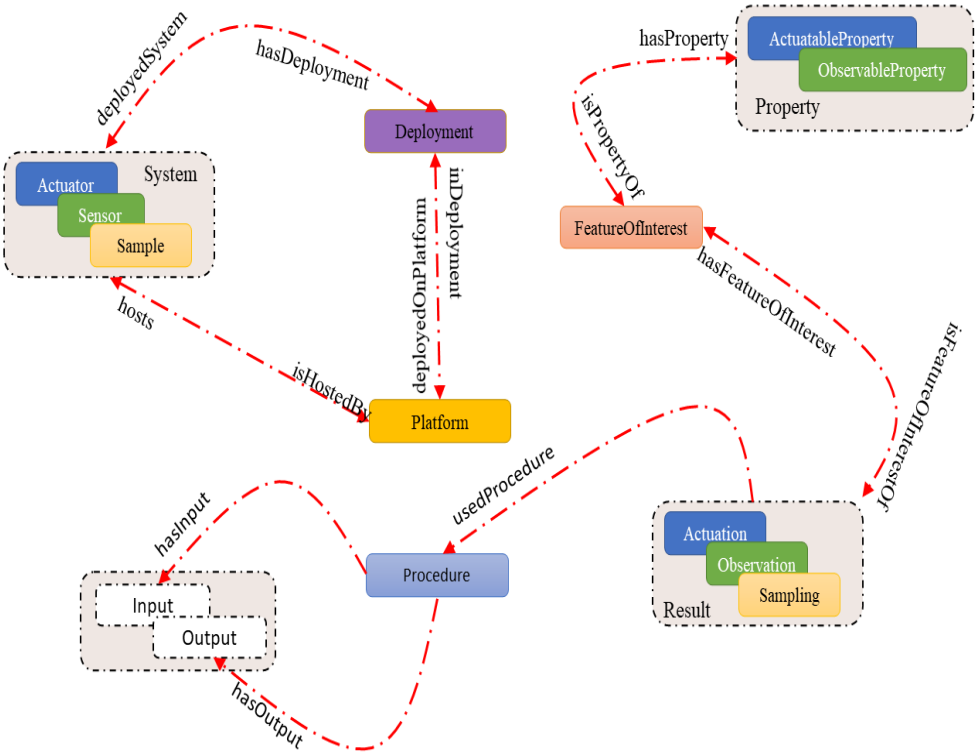


Figure 4.15 An overview of the SOSA/SSN from “sosa:Platform” perspective

In the proposed sensor ontology, there are 4 subclasses of "sosa:Platform" class. These are the prototype types of the sensor nodes created within the scope of the study (SN-A, SN-B, SN-C, SN-D). Within the scope of the thesis, enough individuals have been created from these subclasses.

ssn:System: This class is an abstraction class for ontology concepts that implement procedures. A sensor that operates a procedure that measures the ambient temperature is that can be given as an example of "ssn:System" classes. This Class divide into three subclasses named “sosa:Sensor”, “sosa:Actuator” and “sosa:Sampler”. In the proposed thesis, 5 different sensors and 1 actuator were used. Therefore, 5 subclasses of "sosa:Sensor" class were created.

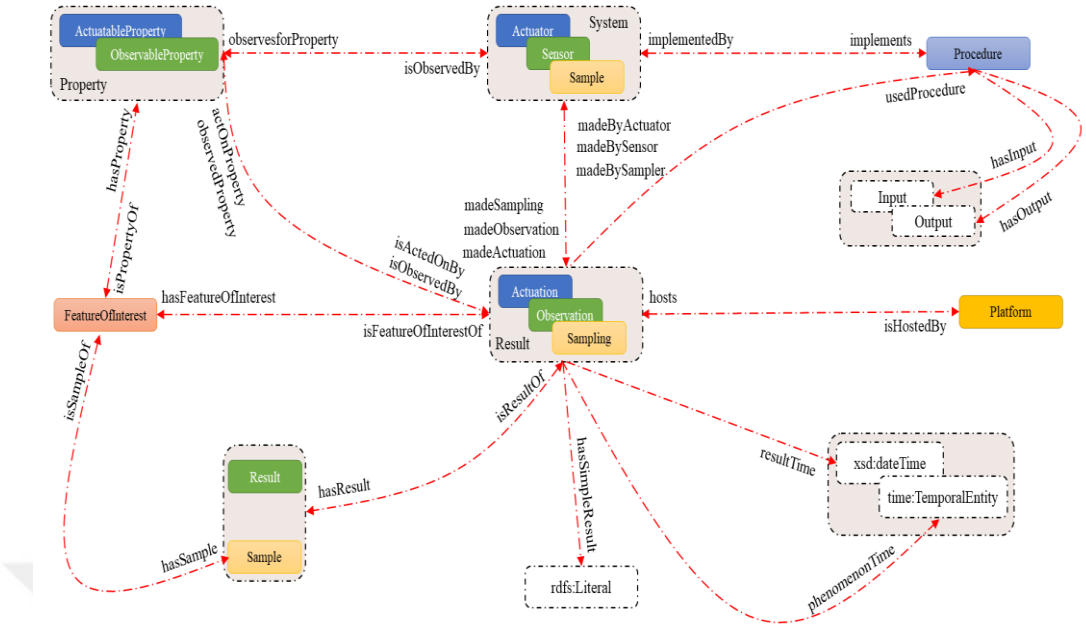


Figure 4.16 An overview of the SOSA/SSN from “sosa:Observation”

sosa:Actuation: In SOSA/SSN ontology, an actuation is the concepts that can change the state of a "sosa:ActuatableProperty" belonging to "sosa: FeatureOfInterest" using "sosa:Procedure". When the temperature in a room falls below a certain degree Celsius, the activity to automatically turn off the ventilation system can be given as an example of the "sosa:Actuation" class. Within the scope of the proposed study, only one actuator has been defined. An overview of the SOSA/SSN classes and properties from "ssn:System", “sosa:Obsevation”, and “sosa:Actuation” perspective (Closely related to these class) is given in Figure 4.16.

sosa:Observation: The "sosa:Observation" class is one of the most important classes of SOSA/SSN ontology. Measures or calculates the value of a "sosa:ObservableProperty" belonging to the selected "sosa:FeatureOfInterest" class using "sosa:Procedure". In other words, it is the result of sensor measurement or actuator action. The action of measuring the change in the concentration of PM10 in the environment with the Nova Pm device can be given as an example of the "sosa:Observation" class. Within the scope of the proposed thesis, individuals of "sosa:Observation" class were added automatically by the control program.

4.4 RDF Triple Database, Data Mining Program, and Ontology Editors

In the proposed thesis, apart from the materials described above, there is a need for an ontology editor to expand the SOSA/SSN ontology for the purpose of the study, an RDF database to save sensor data as a triple, and a data mining program developed to test and compare classical machine learning algorithms in a proactive system design. has been heard. Within the scope of the proposed thesis study, which programs, database, and editor are used and why these applications are selected are explained in detail in the following sections.

4.4.1 Protege Ontology Editor

Today, there are many programs used as ontology editors. Developers' favorite editors include Apollo (Apollo, n.d.), OntoStudio (Weiten, 2009), and Semantic Web Ontology Overview and Perusal (Swoop) (Kalyanpur, Parsia, Sirin, Grau, & Hendler, 2006). This proposed sensor ontology was designed with the Protégé ontology editor developed by Stanford University. Protégé is a free open source framework that provides an interface for users to define ontologies (Musen & Team, 2015).

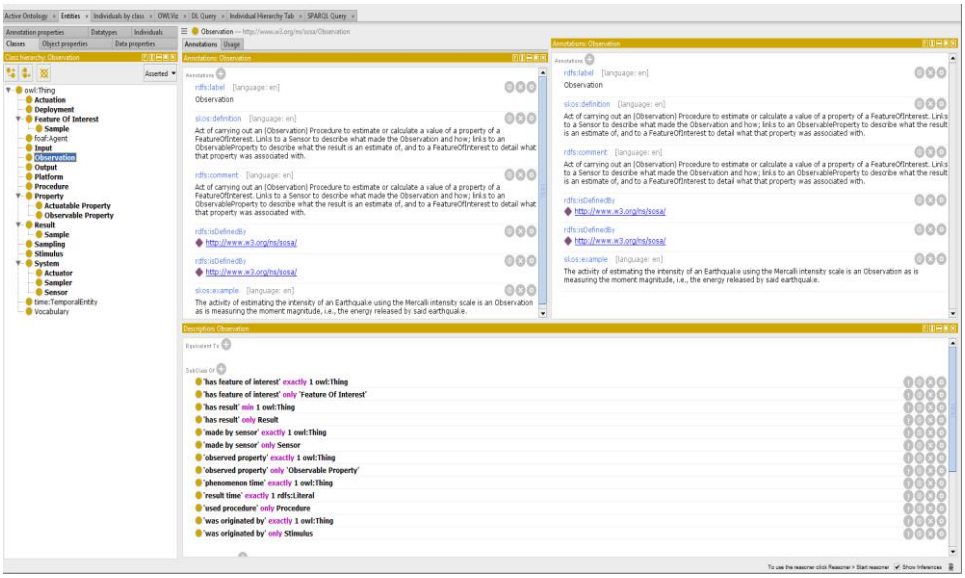


Figure 4.17 The user interface of the protégé ontology creation editor

Protégé 5.5.0 editor has the skills of creating classes and subclasses, defining and visualizing the relationships between classes in order to expand SSN ontology. Protégé's plug-in architecture can be adapted to build both simple and complex ontology-based applications (Stanford Team., 2016). The user interface of the Protégé ontology creation editor is given in Figure 4.17 from the classis's hierarchy perspective.

Within the scope of the proposed thesis, the biggest factor in choosing Protégé as the ontology creation editor is that there is sufficient documentation on the web and it is actively supported by the developers. In addition, Protégé fully supports the W3C's latest OWL 2 Web Ontology Language and RDF specifications. Protégé is a Java-based ontology editor designed to create an extensible and rapid prototype. Developers can integrate the output of Protégé with rule systems or other problem solvers to construct a wide range of intelligent systems (Musen & Team, 2015).

4.4.2 Apache Jena Fuseki

There are RDF triple stores such as AllegroGraph (Graph, n.d.) and Virtuoso (Virtuoso, n.d.), which researchers often prefer in the literature. In the proposed thesis, Apache Jena Fuseki (AJF) was chosen as an RDF triple store. AJF is a SPARQL server. AJF can run as an operating system service, as a Java web application, and as a standalone server (Apache, 2011).

The biggest factor in Apache Jena Fuseki's choice as an RDF triple store is that it works compatible with the DotnetRDF library used in the Controller program. Another reason is that it is a platform the research team is familiar with compared to other Triple Stores. Fuseki is tightly integrated with TDB to provide a robust, transactional persistent storage layer, and incorporates Jena's text query (Apache, 2011). The user interface of the AJF RDF Triple Store is given in Figure 4.18.

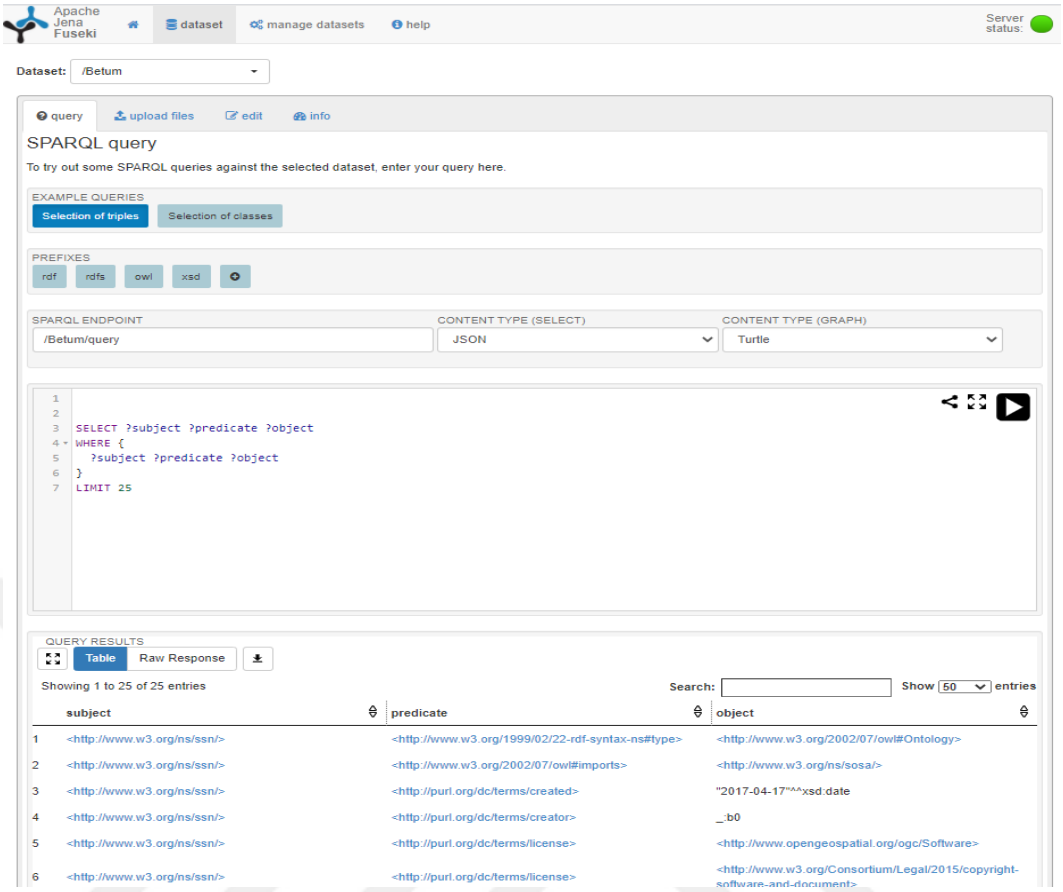


Figure 4.18 The user interface of the Apache Jena Fuseki RDF Triple Store

4.4.3 Rapid Miner

In today's data science, ML methods and DM approaches have been integrated and widespread in many applications and platforms day by day. The spread of these approaches to every field paved the way for potential data mining software and in a short time, many programs were recognized by users and adopted by developers. In the literature, there are many applications such as Weka (WEKA, n.d.), Tableau (Tableau, n.d.), and Knime (KNIME, n.d.) for DM approaches, and ML needs. In the thesis study, the integration of classical ML algorithms into the proposed sensor ontology was done with Rapid Miner data mining software. The design interface of the Rapid Miner development program is given in Figure 4.19.

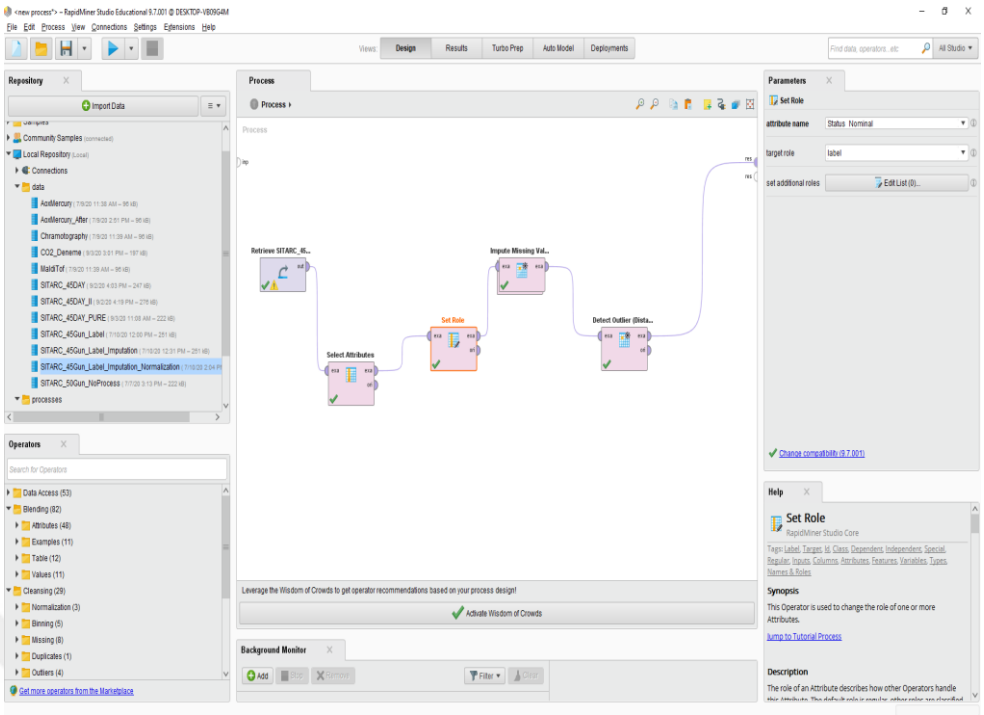


Figure 4.19 The design interface of the Rapid Miner development program

RapidMiner is a client/server architecture-based software platform developed for machine learning and data mining needs. It mainly focuses on research and education. In this sense, it is possible to qualify RapidMiner as a community founded software. It has widespread commercial use as it can also be used for purposes such as rapid prototyping and application development. RapidMiner Studio, RapidMiner Server, RapidMiner Radoop, and RapidMiner Cloud can be used free of charge by members of the community and for academic research.

Within the scope of the proposed thesis, classical machine learning algorithms used for a proactive system design and which data processing methods are implemented using the Rapidminer data platform on sensor data collected.

CHAPTER FIVE
EXPERIMENTAL SETUP

5.1 Overview of This Section

In this part of the proposed thesis, firstly, the design of the sensor nodes, the properties of the WSN installed, and the placement of the sensor nodes in the location determined as the measurement area in SITARC and MICU are explained. In order to create the ontology of the data collected from the sensor nodes, how the SOSA/SSN framework was developed for the proposed thesis study and the criteria by which ontological rules are defined will be detailed. Finally, classical ML algorithms that can be used for a potential proactive system design will be implemented into ontological sensor data and it will be discussed which of these approaches may be more useful on ontological sensor data.

In the next section, comparing the results, it will be discussed that the most appropriate machine learning approach is more effective on ontological sensor data, especially in critical areas such as hospitals, laboratories, and schools. The workflow diagram of the proposed system is given in Figure 5.1. The architectural structure of the proposed Ph.D. thesis is composed of 5 different layers, each of which has its own specific functions and characteristics. These layers are (i) Sensing and WSN Layers, (ii) Semantic Web Processing Layer, (iii) Data Processing Layer, (iv) Decision and Control Layer, (v) Presentation Layer. These layers are explained following briefly.

5.1.1 Sensing and Wireless Sensor Network Layer

This layer represents the environment where the sensor nodes are deployed and the WSN installed. Since the two use cases are considered in this thesis explained in the previous section, two different environments are expected. These use cases are 1- SITARC, and 2- MICU. Sensing and WSNs Layer includes collecting data of the parameters determined from indoor environments selected as measurement environment and transmitting them to the Base station via a Gateway node. The created sensor nodes are distributed in the measurement area in a way that guarantees the

measurement of every parameter for each environment. In other words, the sensors are placed in each independent part in the measurement areas, depending on the size, at least one of each sensor.

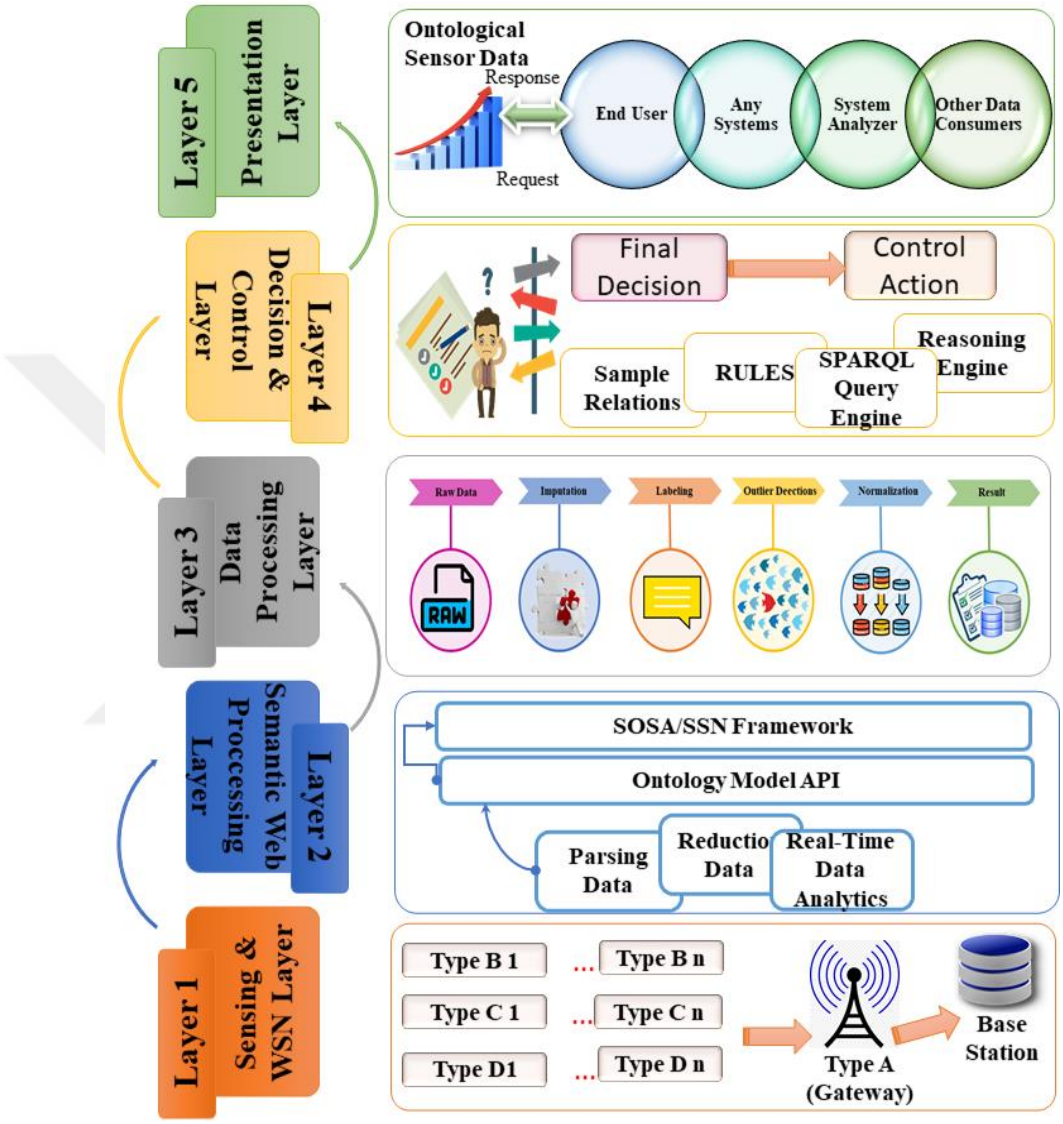


Figure 5.1 Flowchart of The Proposed Thesis Study

5.1.2 Semantic Web Processing Layer

This section is the layer where the sensor ontology begins to be created. The transfer of data to the ontology created using the SOSA/SSN framework takes place in this layer. In this section, first of all, many preliminary procedures such as parsing, cleaning, minimizing, and analyzing data transferred from the previous layer are

applied. Also, all data pre-processed in this layer receive a unique identity for easier access to them (end-user, analyzer, any systems, etc.) consumers.

After all these processes, RDF triples are created by establishing relationships between data. These created triples are saved to the local Apache Jena Fuseki Server with the help of the dotNetRdf library. When saving these triples, the previously expanded SOSA/SSN platform is taken as a metadata framework. Each concept and their relationships that are included in this framework are represented in RDF and XML data format. When necessary, these data stored in Fuseki are taken back to the development environment using SPARQL and transferred to the next layer for processing.

5.1.3 Data Processing Layer

This layer is the first part where raw data from WSN is processed for Machine Learning (ML) approaches. This data, which is transferred to the redevelopment environment using SPARQL, must go through some preliminary processes before being used for a proactive system design. The concept of pre-processing of data includes processes such as cleaning the data, reducing it if necessary, imputation of the data set, determining outliers, and labeling the data. Because the processing of the data sets of ML approaches in appropriate formats and after several operations provide more accurate and consistent results as output.

5.1.4 Decision and Control Layer

One of the most important layers of the proposed thesis project is the Decision and Control Layer. This layer uses the values measured by the sensor nodes to predict the near future. First, the current situation is evaluated. The values from the sensor data are compared with the globally accepted limit values for the measured parameters. If the limit values are exceeded, the necessary control is carried out. Control actions are predefined in the proposed doctoral thesis. Other control actions can be added later in the project as needed.

Control actions correspond to the "sosa:ActuatableProperty" class in the SOSA/SSN framework. Control actions can be evaluated under two main headings. These are the Action and Alert classes. Actions such as operating the air conditioner, opening the vent, or opening the window are considered a member of the "Action" class. On the other hand, giving alerts from system software, and sending SMS to staff is also a member of the Alert class. However, since the regions selected as use-cases are critical regions, the control actions in the Action "sosa:ActuatableProperty" class could not be implemented, since permission could not be obtained.

If the incoming sensor data does not exceed the limit values, historical data is analyzed and the near future estimation is made for some measured data. Two groups of methods were used in the comparison of algorithms in the near future predictions. One of them is statistical methods and the other is ML algorithms. However, this part of the proposed doctoral dissertation will focus on ML methods. There are many ML methods for data analysis and prediction in the literature. However, it is very difficult to determine which method will work best for each event and situation. For this reason, after collecting the data, many classical ML methods were tried and the results were compared to many parameters such as performance, accuracy, and flexibility. These tests were performed by the RapidMiner program and algorithms producing the best result was used as an estimation algorithm in the proposed thesis.

5.1.5 Presentation Layer

This layer is through which sensor data reaches the end-user. In this layer, users can view and analyze data. The presentation and visualization of the data in the system involve analyzing the data coming as a result of the ontological query and presenting it to the data consumer in graphical or list form by using sorting and filtering methods. Especially large data heaps can be very difficult to understand and interpret. Presenting the collected data graphically and as a list makes the data easier to understand for consumers and increases the interaction between the data. Although the user interface in this layer is first thought of as a windows application, it is planned to design web-

based and mobile-based user interfaces in order to enable data consumers to access data from anywhere and from various applications.

5.2 Sensor Nodes Design, and Establishment of WSN

The microprocessor, sensors, and communication device to be used in the study were evaluated from perspectives such as cost, suitability to the project, stability, and accuracy after extensive research and many trials, and the most suitable materials were selected. Within the scope of the proposed thesis, 4 types of sensor nodes were designed to fulfill 4 different tasks. These are named as Type A Node (Gateway Node), Type B Node (Sensor Node 1), Type C Node (Sensor Node 2), and Type D Node (Repeater Node), within the scope of the proposed thesis study. It will be referred to like this in the next part of the thesis. Sensor nodes created are shown in Figure 5.2.

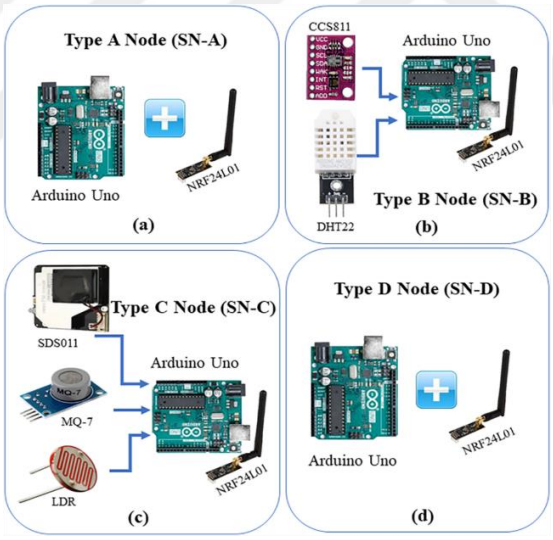


Figure 5.2 Sensor nodes created to collect data from measurement environments

The sensors used within the scope of the thesis study were placed and distributed on two platforms such that the number of parameters is divided into two equal parts. There are two reasons why sensors are divided into different platforms. The first is to reduce the load on the nodes. Another reason is that the sensors can be flexible when they are distributed to the measurement environments. These sensor nodes are designed to collect parameter data in SITARC and MICU that may adversely affect the results of patients, employees, and laboratory analysis.

Type A Sensor Node (Gateway Node): This node is the most important node in the network, as it where all data is collected and transmitted to the base station. In cases where the Type A sensor node fails to function due to physical obstacles or any problem arising from its electronics, or if communication with other nodes is interrupted, all data communication in the network stops. That's why the Type A sensor node is vital. No sensor was placed on it because it did not make any measurements in the environment.

Type B Sensor Node (Sensor Node 1): In the proposed project, 5 different sensors are used to measure 8 parameters. These sensors are integrated into the two nodes, measuring an equal number of parameters.

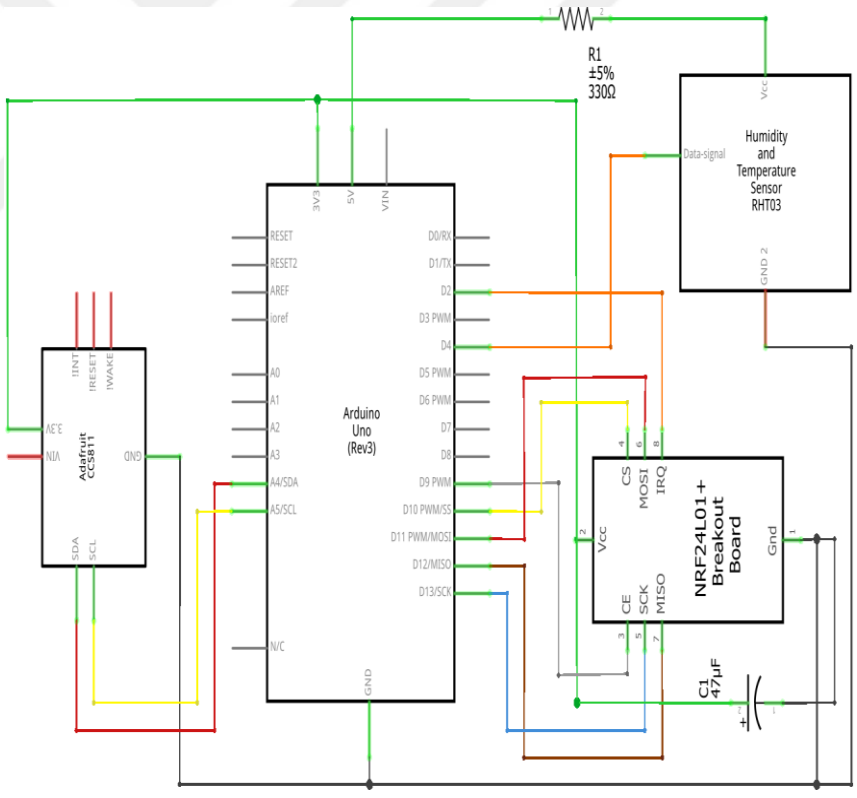


Figure 5.3 Fritzing-drawn circuit modeling of a Type B sensor node

The DHT22 sensor, which measures the temperature and humidity parameters in the environment, and the CCS811 sensor that measure the CO_2 and TVOC, are

integrated on the Type B Sensor Node. The schematic design of the Type B sensor node, prepared with the Fritzing circuit modeling program, is shown in Figure 5.3.

Type C Sensor Node (Sensor Node 2): Another sensor node that makes measurements in the environment specified as the use case for the proposed project is the Type C sensor node. MQ-7 sensor measuring carbon monoxide, Nova SDS011 Sensor measuring PM2.5, and PM10 values, and light-dependent resistance (LDR) sensor measuring light intensity in the environment are integrated into this node. The schematic design of the Type C sensor node, prepared with the Fritzing circuit modeling program, is shown in Figure 5.4.

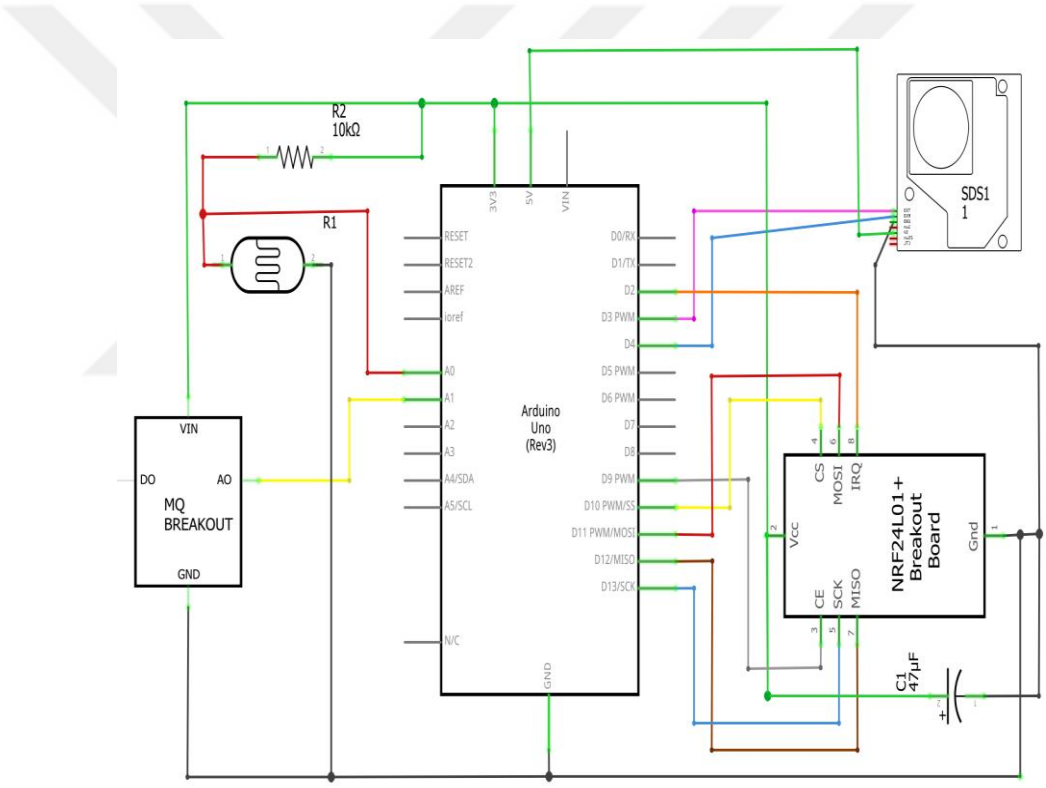


Figure 5.4 Fritzing-drawn circuit modeling of a Type C sensor node

Type D Sensor Node (Repeater Node): After the created nodes were placed in the measurement environment and WSN was established, a communication problem occurred due to the distance and obstacles between some nodes. In order to solve this communication problem and to ensure healthy data communication, repeater nodes were placed that strengthen the received signal and enable the data received from the node to reach the gateway node. The sensors used, the nodes created, the technical

infrastructure of this network, the characteristics, and detailed description of this system used are available in the previous study of the research team (Aktaş, Milli, Lakestani & Milli, 2020).

Arduino Uno was used as a microprocessor in sensor nodes as stated in previous sections. While generating the prototypes of the sensor nodes, Arduino Uno R3 Protoshields were used to get rid of the disadvantages of wiring and modeling on the breadboard. The nRF24L01 + is generally a module sensitive to voltage fluctuations. One of the disadvantages is that there is no light on the module indicating whether it is receiving electricity, so it cannot be determined whether the module is working or not. If there is a voltage fluctuation in the installed system, the nRF24L01 + module does not work and cannot communicate. To solve the communication problem caused by voltage fluctuations and interference between sensor nodes, a capacitor of 10 uF 50 V was placed between nRF24L01 + GND and VCC pins. Thus, after the sensor nodes were distributed in the measurement environment, there was no communication problem between the sensor nodes when there was no obstacle and distance problem.

5.3 Use Cases and Deployment of Sensor Nodes into These Area

2 different domains and environments have been chosen to implement the proposed system as a real-world use case. The first of these are certain laboratories that are actively used in the SITARC of BAIBU and where various physical, chemical, and biological analyzes are performed. Another environment chosen as an example of use is the patient care rooms used in BAIBU the MICU.

5.3.1 Deployment of Sensor Nodes in SITARC Environment

In SITARC, which is selected as a measurement area, there are more than 10 academic laboratories used by academic staff to carry out their analyzes. Within the scope of the proposed thesis study, due to the limited budget, 3 of these laboratories were determined as active measurement areas and sensor nodes were placed. These laboratories determined as measurement areas are MaldiTof, AoxMercury, and Chromatography.

By placing sensor nodes in these average-sized laboratories in a way that there is at least one sensor from each sensor in each laboratory, it is ensured that the 8 parameters to be measured are also measured. For the measurements in SITARC, a total of 1 Type A Node (Gateway), 3 Type B sensor nodes, and 3 Type C sensor nodes were initially designed. The Type B and Type C sensor nodes described in Section 5.2 have been deployed in these 3 laboratories to be used in this case study.

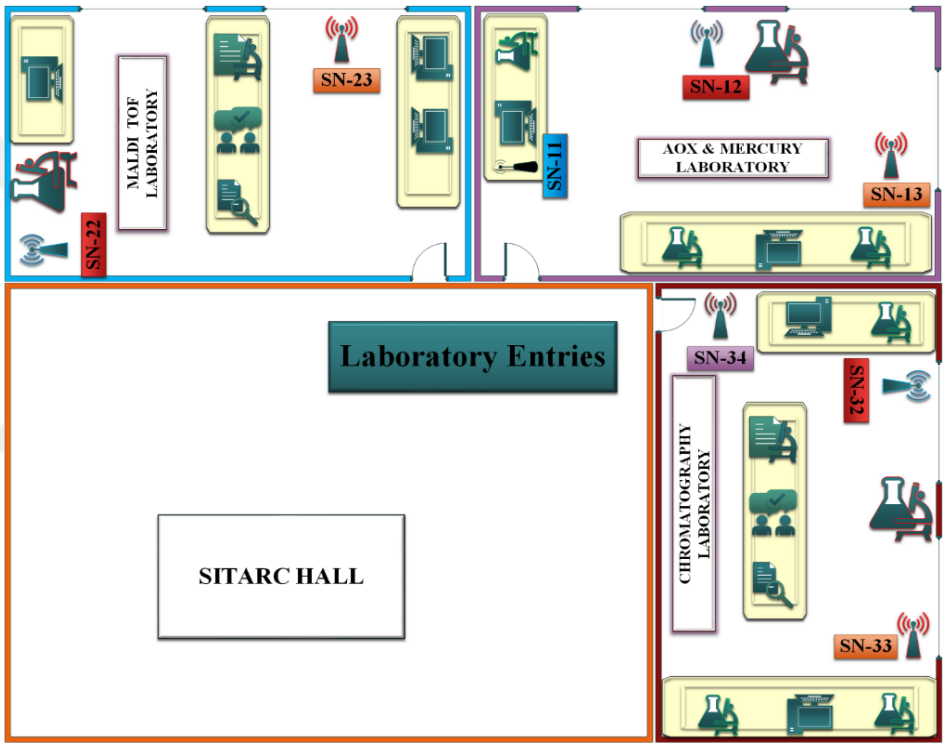


Figure 5.5 Deployment of nodes in the SITARC environment

Since one of the objectives of the proposed study is to determine the IAQ to protect the health of the analyst, it has been deemed appropriate to deploy the sensor nodes at a height of approximately 1.5 meters which is considered as the average breathing level. Besides, these sensor nodes have been deployed near laboratory devices and tubes where the gas density is expected to be high. The deployment of sensor nodes in SITARC to laboratories is given in Figure 5.5.

According to the in Figure 5.5, SN_12 and SN_13 nodes have been placed in the AoxMercury laboratory. SN_22 and SN_23 sensor nodes were placed in the MaldiTof laboratory. Finally, SN_32 and SN_33 sensor nodes were placed in the Chromatography laboratory, and sensor measurements were performed.

The SN_11 Gateway Node has been deployed in the AoxMercury laboratory, which is in the middle of these three laboratories. Sometimes SN_11 in the AoxMercury laboratory and SN_33 nodes in the Chromatography laboratory had problems in communication due to the distance and obstacles. Therefore, only one number of SN_34 has been placed in the Chromatography laboratory in a location close to SN_11. Thus, the interruption of communication between these two sensor nodes was prevented.

Inadequate environmental parameters in buildings such as hospitals, schools, etc. may cause short and long-term health problems such as fatigue, headache, dizziness, respiratory diseases, and cancer in individuals who spend most of their time in buildings. However, inadequate environmental conditions in laboratory environments not only threaten human health but can also significantly affect some analysis results. For example, temperature rise in the Chromatography laboratory significantly affects the performance of PM and gas chromatography devices. In the VOC analysis performed in this laboratory, the increase in the concentration of TVOC in the environment adversely affects the analysis results. Light level, ambient temperature, and CO_2 parameters are effective in the microorganism culture developed in the MaldiTof laboratory.

The number of examples to be given to the effect of the parameters to be measured on laboratory analysis results can be increased within the scope of the proposed study. In addition to these, there are expensive devices such as a spectrophotometer, Maldi Tof/Tof-Ms biotyper system, headspace sampler, thermal desorber, U-Hplc Ecd detector in the laboratories to be measured. Increase CO_2, temperature, and humidity levels in their environment can cause these devices to corrode. This leads to cost losses by revealing the need for maintenance in the devices over time.

While a particular increase in some parameters in the laboratories is positive for human health, it may have adverse effects on the active life of the devices in the laboratory and the results of the laboratory analyses. It is ideal for employees to have a working temperature between 23 °C and 25 °C. However, this increase in temperature causes the organic materials to be deformed more quickly and will directly affect the results of the analysis. For example, when identifying microorganisms in the MaldiTof laboratory, the maximum average temperature should be almost 18 °C.

Otherwise, the culture gets older quickly and causes the results of the analyses to be misleading. Therefore, it becomes more complicated to monitor the parameters in the laboratory and to regulate the appropriate environment in a way that does not threaten human health, does not affect the results of the analysis, and does not shorten the life of the devices. In order to overcome this complex situation, different applications and solutions than classical methods are required. In the case study, the ontology of the sensor data is created, limits and rules are defined to overcome this complex situation.

5.3.2 Deployment of Sensor Nodes in MICU Environment

The second use-case chosen as the measurement area was chosen as the MICU, where most of the patients with impaired vital functions and vital risks were given 24-hour vital support. The measured MICU consists of the main room where the normal intensive care patients stay and the isolated room where the patients with more serious illnesses are monitored. Within the scope of the proposed thesis, sensor knots were not placed in other rooms due to the limited budget. The designed sensor nodes were placed in these two rooms, which are critical for the vital activities of the patients.

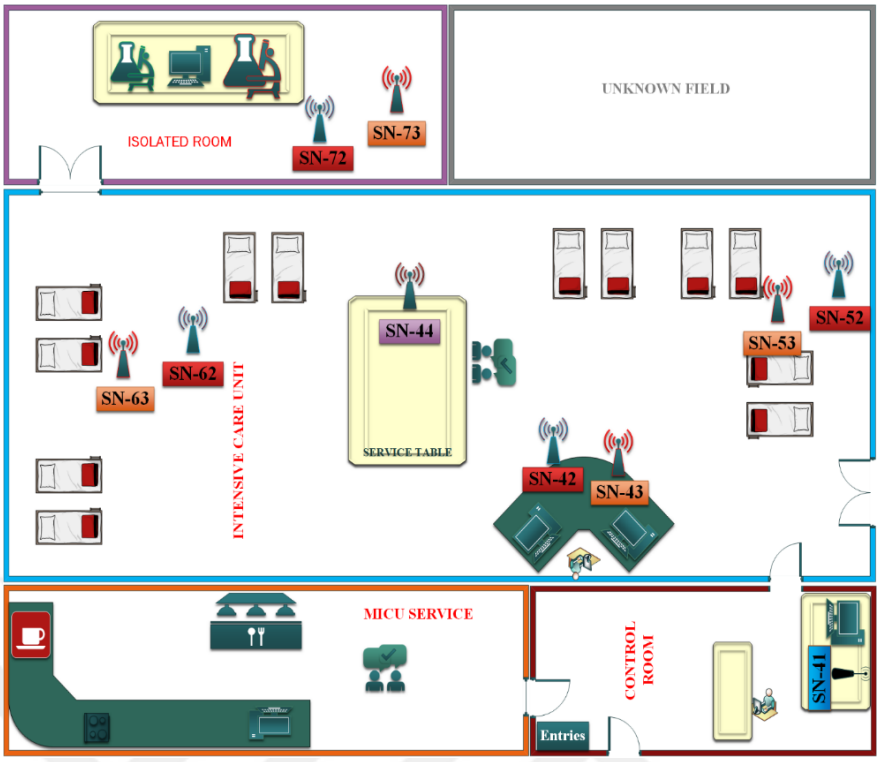


Figure 5.6 Deployment of nodes in the MICU environment

For the MICU use case, a total of 9 sensor nodes were designed, including 1 Type A node, 4 Type B Sensor nodes, and 4 Type C Sensor nodes. However, after the sensor nodes are created and distributed to the MICU, it has been observed that Type B and Type C nodes placed in the isolated room sometimes experience problems in communication due to the distance and physical barriers to the Gateway node Type A node. Therefore, only one number of SN-D has been placed on the service table which is in the middle of the MICU main room, so that the communication between these sensors did not break and the data transfer continued. Deployment of nodes in the MICU which is selected as a measurement environment is given in Figure 5.6.

In MICU, sensor nodes are positioned around 1.5 meters, which is the human respiratory level, just like in SITARC. The node feeds were provided from the care units at the head of the patient beds, that is, from the mains electricity via an adapter. As can be seen in Figure 5.6, care has been taken to distribute the sensor nodes homogeneously to the environment in order to increase the effective area more. SN_41 sensor node has been deployed in the Control room where the computer is located, in

order to avoid confusion during the performance of the controls required to evaluate the data. Since the control room is outside the measurement area, Type B and Type C sensor nodes are not placed in this room.

SN_42 and SN_43 sensor nodes were located in the patient reception table where there is a lot of intensity, especially during the morning control hours. SN_52 and SN_53 sensor nodes were placed between the inputs in the MICU main room, which has two external inputs. SN_62 and SN_63 sensor nodes were located near the patient beds opposite the entrances. Finally, SN_72 and SN_73 sensor nodes were located in an Isolated room, where patients with more critical conditions were monitored.

Sensor measurements and wireless communication in WSN were checked for a few days after the sensor nodes were placed in the measurement environment. After it was understood that there was no problem in sensors and communication, the data collection process, which was expected to be an important and long process for the thesis study, was initiated. While the data collection process was continuing, the WSN system, which was set up periodically, was checked and, if any, its problems were resolved.

5.4 Collecting Raw Sensor Data

Since the project budget was insufficient to design 18 separate sensor nodes, the measurements made in SITARC and MICU, which were selected as the measurement area, were carried out at different date intervals. Thus, within the scope of the proposed thesis, a total of 10 different sensor nodes were designed, including 1 Type A Gateway Node, 4 Type Sensor Node, 4 Type C Sensor Node and 1 Type D Repeater Node. These designed sensor nodes were first placed in SITARC as described in section 5.3.1 and data collection was performed. When the data collection process in SITARC was finished, the sensor nodes were collected and deployed to MICU as described in section 5.3.2 and the data collection process was performed there. When, how, and how the data are collected and stored in SITARC and MICU selected as measurement areas are explained in detail below.

5.4.1 Collecting Raw Sensor Data in SITARC

After the sensor nodes were placed in the 3 laboratories of SITARC, and the data was sent properly, the data collection process was started on 29.08.2019 at 16:05. Each sensor in the installed system is programmed to measure an average per minute and send it to the gateway. The hourly average of the collected data was added to Apache Jena Fuseki, which is frequently used as a triple database. Jena Fuseki is a SPARQL server. Besides, it has been preferred as a triple database in this project as it provides a clear user interface for server monitoring and management.

The data collection process has been terminated on 12.10.2019 due to the annual maintenance of the devices in the laboratory. A total of 45 days of uninterrupted data was collected at the selected measurement sites. Between these dates, each sensor made approximately 65,000 measurements, and a total of approximately 1,500,000 measurements were made. Theoretical and practical training was given twice in the first 10 days of September and October in the laboratories specified between the dates of measurement, and it was frequently used in 3 laboratories where the measurement was made. This situation has been beneficial for the project results in terms of seeing what kind of changes may occur in the parameters during the analysis and training in the laboratory.

5.4.2 Collect Raw Sensor Data in MICU

After the sensor nodes were placed in the 4 locations in MICU, and the data was sent properly, the data collection process was started on 02.03.2020 at 11:03. Each sensor in the installed system is programmed to measure an average per minute and send it to the gateway like at SITARC. The hourly average of the collected data was added to Apache Jena Fuseki like the data gathering process in SITARC. Jena is a SPARQL server and frequently used as a triple database by many researchers. In addition, Fuseki has been preferred as a triple database in this project as it provides a clear user interface for server monitoring and management as mentioned before.

The data collection process carried out at MICU was completed on 01.04.2020 due to the measures taken within the scope of the Covid-19 outbreak, which was effective across the world. A total of 30 days of uninterrupted data was collected at the 4 different selected measurement sites. Between these dates, each sensor made approximately 43,000 measurements, and a total of approximately 1,400,000 measurements were made. Within the scope of this project, in the MICU designated as the second measurement area, doctor controls were carried out between 09:00 and 11:00 am every day. In addition, every day between 12:00 and 14:00 is determined as visitor hours. These situations have been beneficial in terms of evaluating the results of the project to what extent the human density and human activities affect the environmental conditions during these hours.

5.5 The Controller Program Design

The user interface designed within the scope of the proposed project can be used for data processing, saving, editing, visualization, listing, etc. It is designed to perform many operations such as. Its most important function parses the sensor data coming from WSNs, saves it to the database, and manages it. Apart from that, the incoming sensor data are shown in the interface as both historical data and real-time. In addition, data can be filtered according to sensor node number and sensor type or parameter. At the top of the user interface, the last data from sensor nodes and their time are displayed. This is intended to be immediately detected and reacted when there is a delay in any sensor node.

In the middle of the user interface, the graph of the sensor data is plotted real-time. However, this section has been canceled during the data collection phase because the desired efficiency cannot be obtained from the graphic due to the abundance of sensor data and the system has to work for a long time. The edited data are created in the form of RDF triples in the background and saved in the RDF database by providing a connection to the Fuseki Server database. The Control Unit, which was designed within the scope of the thesis study, is given in Figure 5.7.

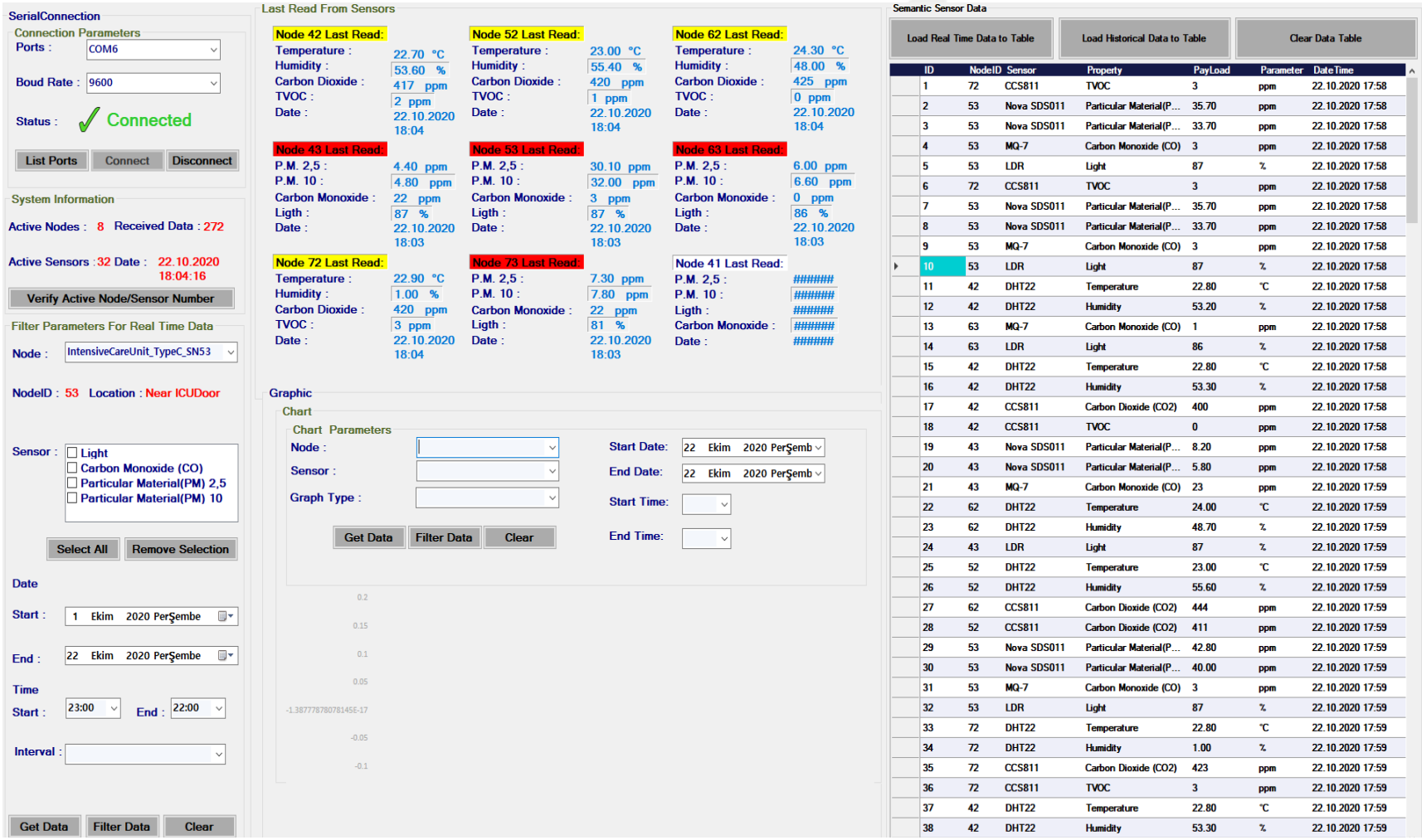


Figure 5.7 Program interface of the proposed study

5.6 Ontology Development Process

In the scope of this study, an example of SOSA/SSN, the framework designed by W3C for semantic sensor networks, was generated using data collected from the SITARC and MICU. In this section, how the SSN framework is developed with special concepts and examples and how it becomes suitable for the purpose will be presented. The proposed ontology includes classes, individuals, rules, and their relationships in laboratory parameters monitoring systems. This proposed ontology framework was designed with the Protégé ontology editor developed by Stanford University. Protege is a free open-source framework that provides an interface for users to define ontologies. Protege 5.5 editor has the skills of creating classes and subclasses, defining and visualizing the relationships between classes in order to expand SSN ontology.

The SSN is an application-independent framework which needs to be expanded with specific notion and examples. This expansion process includes has been made by adding some classes, object property, data property subclasses, and individuals that fundamental of ontologies to the SSN core ontology. Some subclasses and individuals added to the SSN/SOSA core ontology are explained as follows. Firstly, the Laboratory and Hospital classes were added as a subclass of the "sosa:FeatureOfInterest" class. In the laboratory environments, since the case study was implemented in 3 different laboratories, there are 3 different individuals of this Laboratory subclass.

These are MaldiTof laboratory, AoxMercury laboratory, and Chromatography laboratory. On the other hand, in hospital environments, since the case study was implemented in 2 different rooms, there are 2 different individuals of this hospital subclass. These are "IntensiveCareUnit" and "ControlRoom". Extended and developed SSN ontology is given in Figure 5.7 from "sosa:FeatureOfInterest" class perspective. The proposed study has been integrated into the 2 different areas described above as a real-world use scenario. When this ontology is desired to be expanded in terms of its usage area in future studies, the environment to be observed

such as a university, school, factory, workplace, etc. should be added to the "sosa:FeatureOfInterest" subclass.

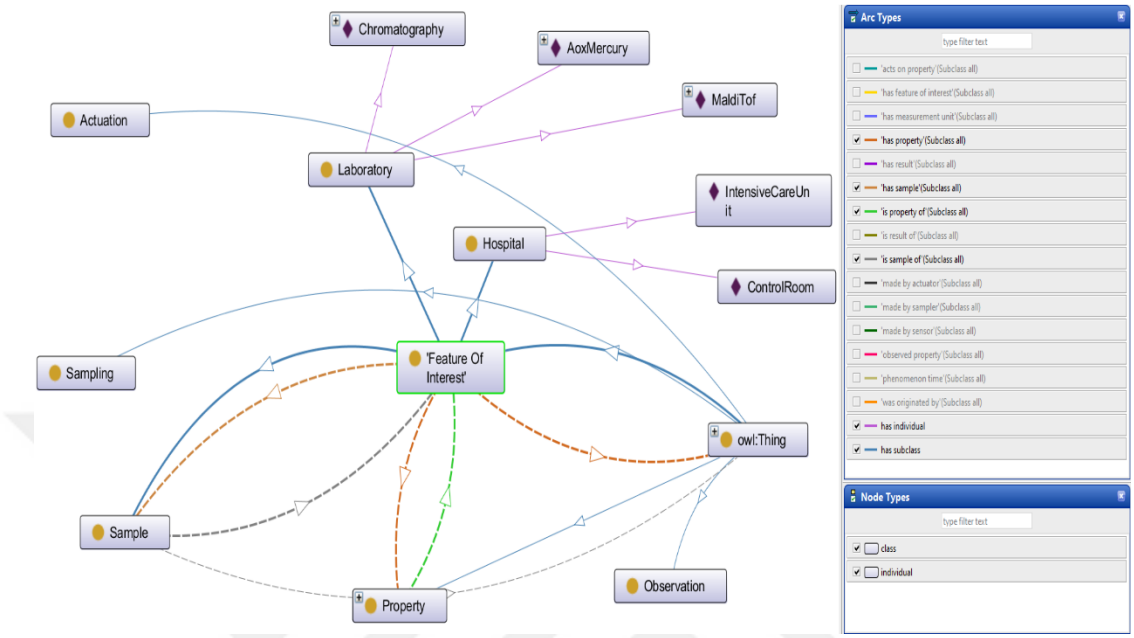


Figure 5.8 Proposed sensor ontology from “sosa:FeatureOfInterest” perspective

Sensor nodes created previously in different types were added as a subclass of the “sosa:Platform” class, which is a concept that includes the standard classes of SSN ontology and other assets, especially sensors, actuators, samplers. In the proposed ontology, since there are 4 different sensor node types, there are 4 subclasses in “sosa:Platform” class. These are SN-A, SN-B, SN-C, SN-D. Since there are two SN-A in the proposed project, individuals of this class are AoxMercurySN11, and IntensiveCareUnitSN41. Extended and developed SSN ontology is given in Figure 5.8 from “sosa:Platform” class perspective.

Within the scope of the proposed project, a total of 7 individuals of the SN-B class, a total of 7 individuals of the SN-C class, and a total of 2 members of the SN-D class were created as seen in Figure 5.9. If a node to the project later to perform a different task will be used, simply this node must be added as a subclass to the "sosa:Platform" class. However, if an extra node is required to perform the same task, it is necessary to add it as an individual of the relevant node.

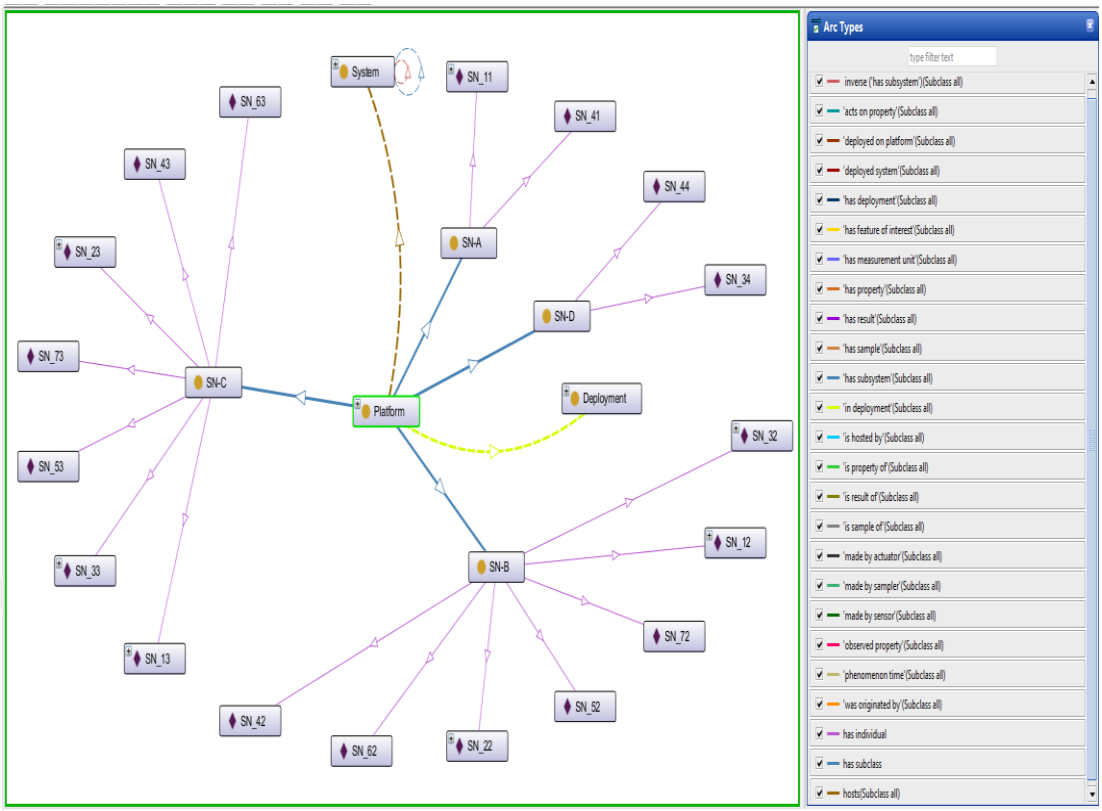


Figure 5.9 Proposed sensor ontology from “sosa:Platform” class perspective

In the class “ssn:System” of core SOSA/SSN ontology already has classes “sosa:Actuator”, “sosa:Sensor”, and “sosa:Sampler”. In the proposed project, within the scope of the extension of SSN ontology, 5 sensors and 1 actuator as described in Section 4.1.3 were added as subclasses.

Each sensor used in the sensor classes has been named using its own name and the number of the node where it is deployed. For example, for the SITARC use case, the DHT22 sensor in the MaldiTof laboratory is named DHT22_22. Likewise, in the MICU use case, the MQ-7 sensor module deployed in “ControlRoom” is named “MQ7_73”. Extended and developed SSN ontology is given in Figure 5.10 from “ssn:System” class perspective.

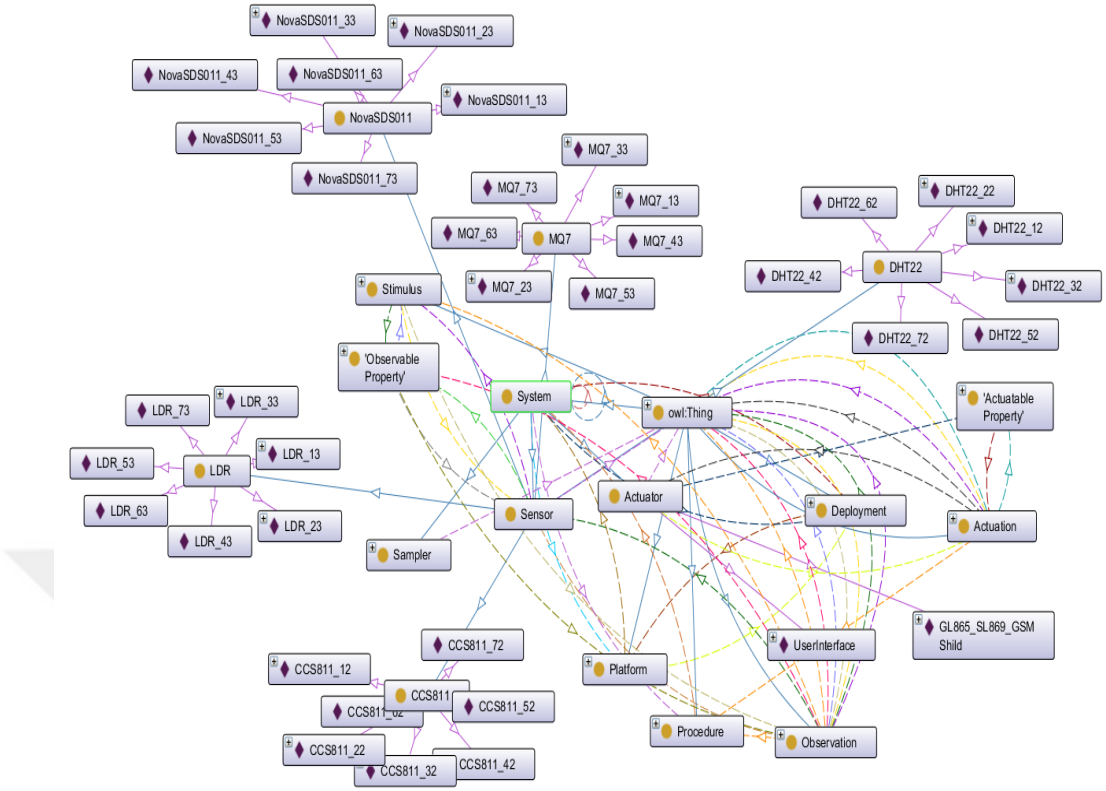


Figure 5.10 Proposed sensor ontology from “ssn:System” class perspective

One of the SOSA/SSN core classes, “sosa:Property”, has 2 subclasses. These are “sosa:ActuatableProperty” and “sosa:ObservableProperty”. These properties in the ontology have been separated and grouped according to their observability and actuatability. Actions and alerts are given as an example of “sosa:ActuatableProperty” subclass.

Temperature, Humidity, CO_2, TVOC, $PM_{2.5}$, PM_{10}, CO, Light that are parameters of wanted to measurement are added to extended SOSA/SSN ontology as individuals of “sosa:ObservableProperty” class. When another parameter is wanted to be observed and activated, it must be added to the related subclass of “sosa:Property”. Extended and developed SSN ontology is given in Figure 5.11 from “sosa:Property” class perspective.

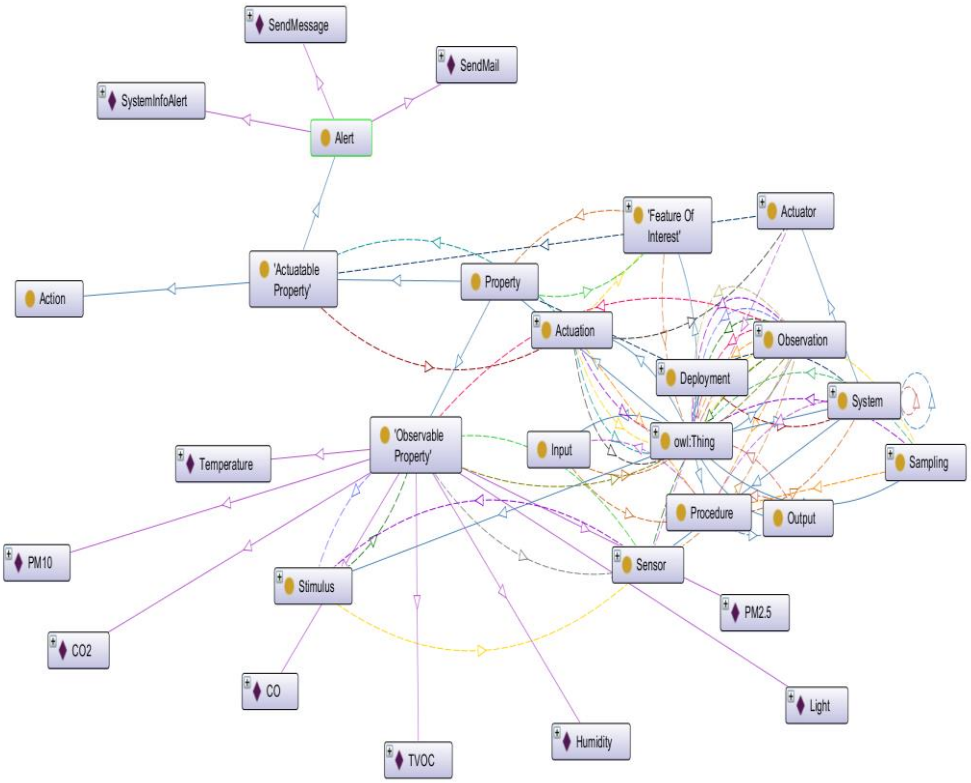


Figure 5.11 Proposed sensor ontology from "sosa:Property" class perspective

There is no "MeasurementUnit" class in the basic SOSA/SSN ontology. The "MeasurementUnit" class was created to avoid unit complexity at the proposed ontology. This class specifies the unit of the "sosa:hasSimpleResult" value of the individuals of the measured "sosa:Observation" class. Within the scope of the proposed project, the units of the parameters measured in laboratories were added as individuals of the "MeasurementUnit" class.

Parts per million (ppm) was used as the unit of measurement for $PM_{2.5}$, PM_{10}, CO_2, and CO. While Celsius was used as the measurement unit for temperature, parts per billion (ppb) was used as the measurement unit for TVOC. Finally, percent was used as the unit of measurement light and humidity. Extended and developed SSN ontology is given in Figure 5.12 from "MeasurementUnit" class perspective.

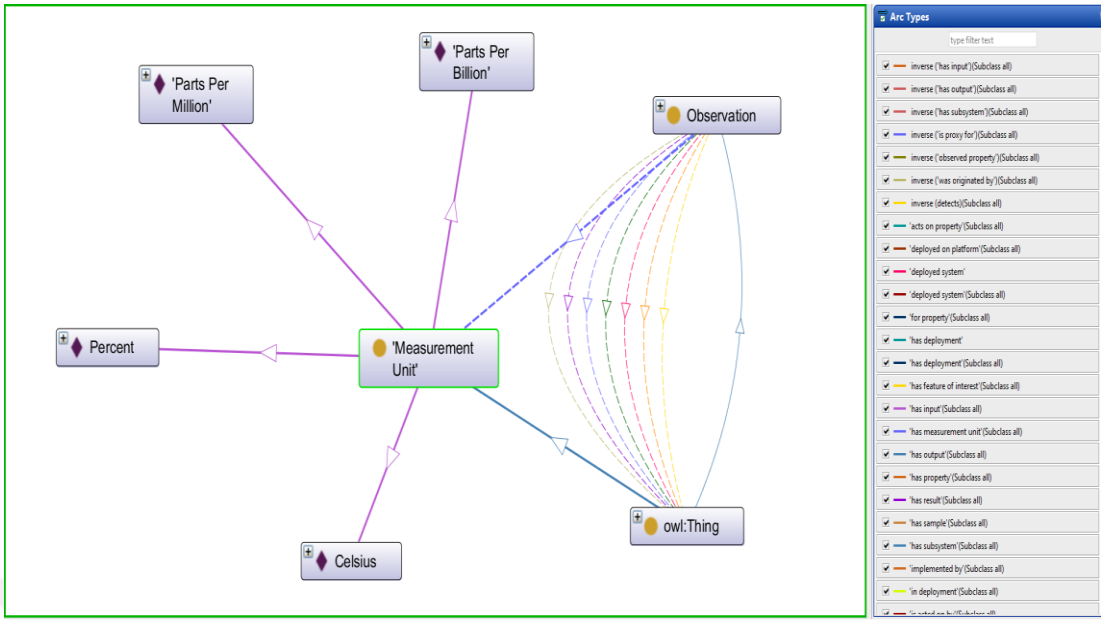


Figure 5.12 Proposed sensor ontology from "MeasurementUnit" class perspective

If other units will be used in different projects, it is enough to add them to the "MeasurementUnit" class. For example, if the temperature is to be measured in Kelvin, it should be added to the "MeasurementUnit" class of the Kelvin unit. Thus, it is considered that there will be no unit confusion between the values to be measured in different projects to be managed under the same framework. Since "MeasurementUnit" is the unit of the observed property, "hasMeasurementUnit" object property has been created between "sosa:Observation" class.

Finally, the most important class for the proposed ontology among these classes is the "sosa:Observation" class. In this study, it was not necessary to add any subclasses for this class. However, each value measured by the sensor data is recorded as an individual in the observation class by assigning a different id of 32 characters. In this way, each observation is ensured that the individual has a unique identity. This allows data consumers to access each observation data directly. Each observation has 2 data properties called "sosa:hasSimpleResult" and "sosa:resultTime". The "sosa:hasSimpleResult" property is the simple value of the "sosa:Observation", "sosa:Actuation" or "sosa:Sampling" action. The "sosa:resultTime" data property shows the time when the "sosa:Observation", "sosa:Actuation" or "sosa:Sampling"

action is completed. Extended and developed SSN ontology is given in Figure 5.13 from “sosa:Observation” class perspective.

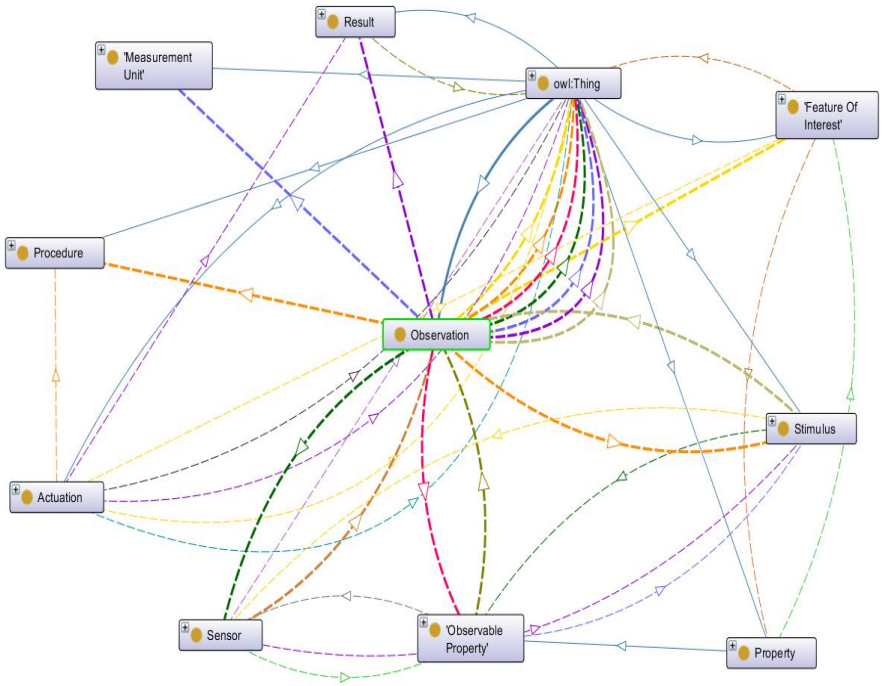


Figure 5.13 Proposed sensor ontology from “sosa:Observation” class perspective

In the proposed thesis, the changes made within the scope of the original SOSA/SSN ontology framework, the added classes, object property, data properties, and instances are explained in general. When the original SOSA, SSN, and the proposed expanded framework are analyzed numerically after the addition process;

There are 16 classes in the original SOSA framework, and 23 classes in the original SSN framework, while 14 more classes were added in the expanded framework and there are 37 classes in total. While there are 21 object properties and 2 data properties in the SOSA frame, there are 36 object properties and 2 data properties in the SSN frame. There are 37 object properties and 2 data properties in total in the extended Sensor ontology framework which is proposed within the scope of this thesis. While there is no individual in the basic SOSA and SSN frames, within the proposed sensor ontology framework, 100 individuals were added and expanded.

5.7 Integrating of ML Algorithms on Ontological Sensor Data

In the Experimental Setup section, sensor data were collected in designated indoor environments as the first step. Later, these sensor data collected were added individually to the Observation Class in the extended SOSA/SSN framework in the Apache Jena Fuseki RDF database. As a final step, in this section, ML algorithms and data mining approaches have been integrated into the data extracted from the RDF database. In this section, many classical ML algorithms have been tried on ontological sensor data for a proactive system design. The results of these tried algorithms were compared in many ways and it was presented which algorithms could be used in a proactive system design for the proposed ontological sensor system, and the results were shared with the academic community.

5.7.1 Pre-Processing of Ontological Sensor Data

For a proactive system design, when ML approaches are used, the data created must first be prepared for ML algorithms. Preprocessing the data set to be used will likely improve performance in most of the implemented algorithms. In this study, ontological sensor data includes preprocessing operations, Data Labeling, Imputations, Outlier Detection, and Normalization processes. These operations applied to ontological sensor data are explained in detail below. The sequence of pre-processing operations performed before implementing ML algorithms on the raw sensor data collected from SITARC and MICU is given in Figure 5.14.

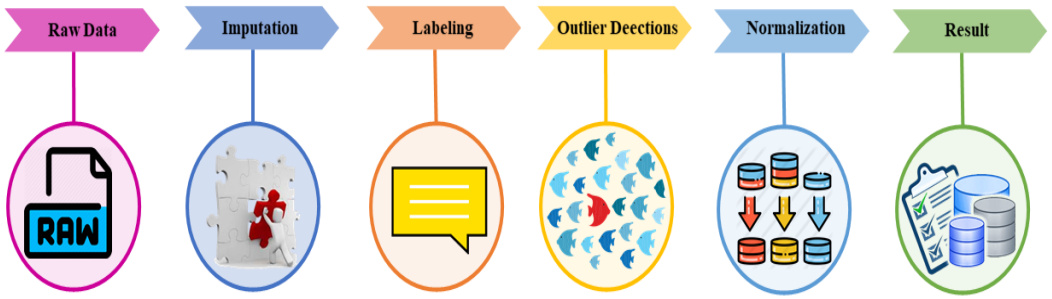


Figure 5.14 The flowchart of implementation in data preprocessing

5.7.1.1 Missing Data Imputation

The data collection phase of the proposed thesis study took 45 days in the first use-case, SITARC. Data acquisition in SITARC was carried out with 15 sensors placed on 8 sensor nodes. As mentioned earlier in this study, 8 different indoor environment parameters were collected in 3 different laboratories in SITARC. Sensor nodes are programmed to measure from each sensor every minute and transmit to the Gateway node. Therefore, in the SITARC use case, a total of 24 measurements are made per minute and given as input to the system.

Approximately 1440 measurement data per hour were taken from the deployed sensors. According to this calculation, 34.560 measurements were made with the help of sensors per day at SITARC. It is expected that approximately 1.555.200 sensor data will be obtained as a result of the 45-day measurement made in SITARC. While these collected data are stored in the RDF database, their hourly averages are recorded so that the data table is not overloaded. Because too much change in minute data is not expected and unnecessary repetitive data may cause the system to slow down.

As all these calculations indicate, when the measurements are completed in the laboratories with SITARC, it is expected that there will be approximately 25,920 data in the database. However, the total data obtained after 45 days is 23,252 due to the malfunction of the devices operating in the system or human error. This number corresponds to approximately 89.7% of the data that should be recorded.

It is important to fill in missing values with a reasonable approach, especially if approaches sensitive to missing values such as "Decision Tree" and "Random Forest" are to be studied. For this reason, data continuity was ensured by filling the 10.3% portion that could not be recorded, using well-known and accepted methods in the literature. In data mining, it is possible to solve the missing value problem with different approaches, for example deleting missing values, accepting them as the average of that feature, or accepting zero are some of the most common missing value solutions.

Deleting or statistically filling missing values may cause bias and negatively affect the result. Therefore, unlike these approaches, imputations of the data can significantly improve the quality of the data set. Recently, many studies have shown that imputing missing values with classification approaches have positive effects on outcomes. Comprehensive information on this topic is given in section 2.4.2.5. Within the scope of the study, the missing values in the SITARC dataset were filled by using the K-NN algorithm and Gradient Boosted Trees approaches together as a hybrid and the quality of the data set was increased. Data imputation was carried out with the RapidMiner data processing program. How the missing values in SITARC dataset are filled using RapidMiner is given in Figure 5.15.

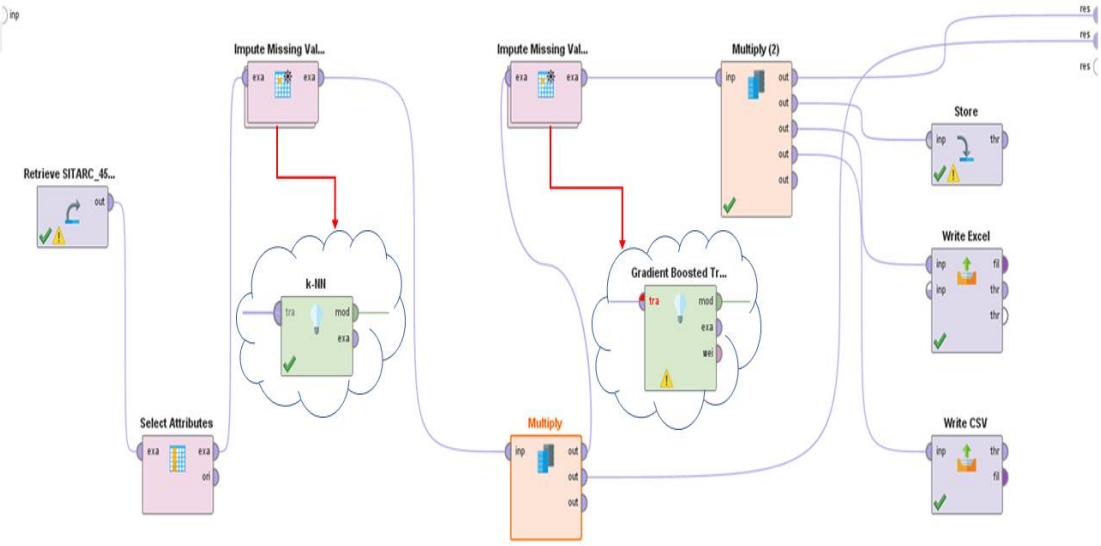


Figure 5.15 Imputation processing in SITARC with RapidMiner

The data collection phase of the proposed thesis study took 30 days in the second use-case, MICU. Data acquisition in the MICU was carried out with 20 sensors placed in a total of 10 sensor nodes. In this study, 8 parameters were measured in 2 different environments in MICU as mentioned before. Sensor nodes are programmed in MICU to measure from each sensor every minute and transmit to the Gateway node, just like SITARC. Therefore, in the MICU use case, a total of 32 measurements per minute are made and given as input to the system.

Approximately 1920 measurements were made per hour from the deployed sensors. According to this calculation, 46,080 measurements were made with the help of

sensors per day in MICU. It is expected that approximately 1,382,400 measurements will have been made after 30 days of measurement at the MICU. While these collected data are saved to the RDF database, their hourly averages are recorded so that the data table is not overloaded. Because too much change in minute data is not expected and unnecessary repetitive data may cause the system to slow down.

As all these calculations indicate, when the measurements are completed in the two environments in the MICU, it is expected that there will be approximately 23,040 data in the database. However, the total data obtained at the end of 30 days is 20,355 due to the occasional malfunction of the devices operating in the system, power failure, or human error. This number corresponds to approximately 88.35% of the data to be recorded. Filling the missing values in the data created by the data collected in MICU with a reasonable approach will increase the quality of the prediction to be made in the future. For this reason, the continuity of data was ensured by filling 11.65% of the unrecorded part with well-known and accepted methods in the literature.

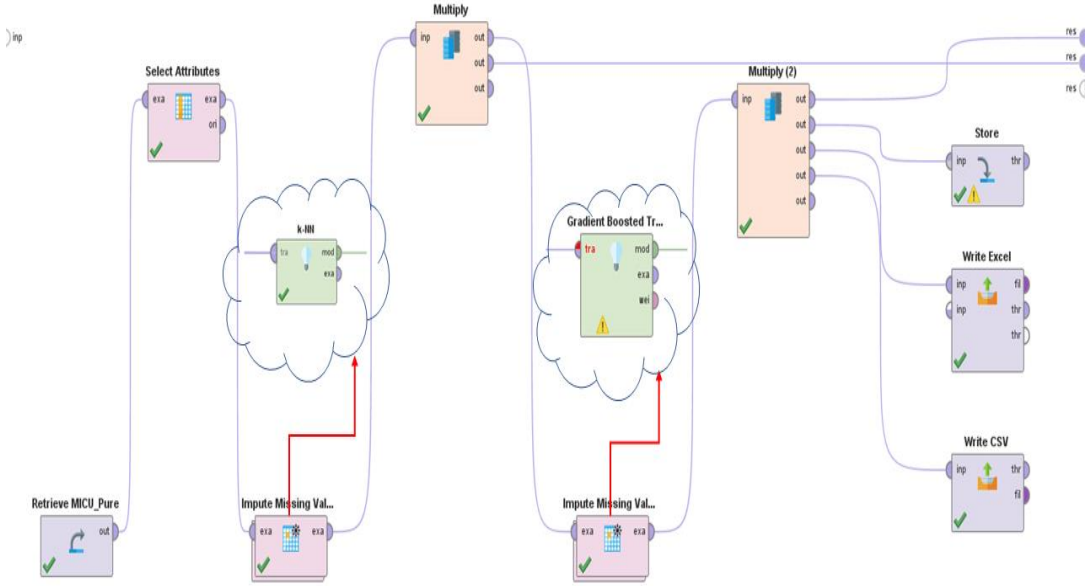


Figure 5.16 Imputation processing of missing values in MICU with RapidMiner

The data collected in MICU within the scope of the study were filled with a hybrid approach as in the first use-case SITARC. This hybrid approach includes the K-NN algorithm and the Gradient Boosted Trees algorithms. Data imputation operation was

carried out with the RapidMiner data processing program. In Figure 5.16, it is schematized how the missing values in the data set consisting of the data collected in MICU are filled using RapidMiner.

As seen in Figure 5.15 and Figure 5.16, firstly the attributes (Columns) that need to be imputed in both data are selected with the "Select Attribute" component in Rapid Miner Studio. The missing values in the data set consisting of the data collected in both use-cases were filled by adding two Impute "Missing Value" components. Two different approaches, namely K-NN and Gradient Boosted Tree, were used with these Impute Missing Value Components, respectively.

Considering the time series data of the K-NN approach, it makes imputations with a high accuracy rate. However, the number of consecutive missing values in some parts of the data sets created may be higher than the value that is used for neighborhood value. Therefore, a hybrid approach has been used to fill all data in the data set consistently. Finally, the new datasets created as a result of the imputation process were saved to Local Repositor with the "Store" component to be used in later operations. The results are also saved in different formats for different purposes with the "WriteExcel" and "WriteCvs" components provided by the Rapid Miner studio.

5.7.1.2 Data Labeling Process

The accepted reference values of important parameters that determine indoor air quality such as CO_2, CO, TVOC, $PM_{2.5}$, PM_{10} have been determined by the institutions that are accepted worldwide such as WHO, EPA, ASHARE in the literature. In this study, these reference values are used while classifying and labeling the data. However, while determining the limit values of parameters such as temperature and humidity, the past experiences of researchers who made analyses in other research and laboratories were used. Although the light level, which is the last parameter measured, is effective in many laboratory processes such as bacterial growth, an accepted limit value has not been found in indoor air quality literature.

According to the WHO, the daily average max values that can be exposed for $PM_{2.5}$ and PM_{10}, which 2 of the measured parameters in the scope of the proposed study, are 25 ppm and 50 ppm, respectively (Krzyzanowski & Cohen, 2008). Studies have shown that exposure to $PM_{2.5}$ and PM_{10} causes respiratory diseases (H. Qiu et al., 2012; Westphal et al., 2013). Moreover, toxicological and epidemiological studies show that $PM_{2.5}$ is particularly harmful because smaller particles are more likely to penetrate deeper into the lungs (Feng, Li, Sun, Zhang, & Wang, 2016; Janssen, Fischer, Marra, Ameling, & Cassee, 2013; Strandberg-larsen et al., 2016).

CO_2 is a colorless, odorless, noncombustible gas that occurs naturally in the atmosphere. Outdoor CO_2 levels generally range from 350 to 400 ppm. According to the WHO, the maximum CO_2 level should be 1,000 ppm for human health indoors (Krawczyk, Rodero, Gładyszewska-Fiedoruk, & Gajewski, 2016). On the other hand, according to the ASHRAE, the maximum CO_2 value in indoor areas should be 700 ppm for humans (Stanke et al., 2007). Ventilation is probably insufficient when the CO_2 level exceeds the reference value, and people often complain of headache, nose and throat discomfort, fatigue, lack of concentration, coma (Hussin, Ismail, & Ahmad, 2017). Since laboratory work continues for long hours, the analysts will likely be exposed to high levels of CO_2 for a long time. In order to prevent or minimize the complaints of people who have to work in laboratory environments, the maximum CO_2 level that ASHRAE and WHO consider appropriate is selected as the limit in this study.

Another parameter measured in this study is TVOC. They are toxins and chemicals that can harm the environment and human health. Health effects can range from minor eye, nose, and throat irritations to liver and kidney damage or cancer, depending on the level of exposure (Zahangeer, Armin, Haque, Halsey, & Qayum, 2018). According to Brown, the average hourly TVOC level is a maximum of 500 ppb (Brown, 2008). CO is a colorless, non-irritating, odorless, and tasteless toxic gas. The average hourly maximum CO level set by the WHO is 35 ppm (World Health Organization[WHO], 2010). In the case of overexposure above the limit CO levels determined by the WHO,

CO poisoning occurs. CO poisoning causes serious problems from headaches, nausea, and vomiting to cardiac arrest, respiratory arrest, and coma (Wilbur et al., 2012).

One of the most critical factors affecting the performance of the device in mass measurement analysis is temperature. If the temperature is outside the limit values, it may cause undesirable conditions in the analysis results. Therefore, the average hourly temperature was taken between 18 °C to 22 °C in order to minimize error from the analysis results in the laboratories where the measurement was performed. Humidity in the environment causes the devices to rust quickly and shorten their life. For this reason, the humidity limit values in the environment to be measured should be between 35%-70%, which is the limit values for human health. A very high light level will cause the aging of the sample to be studied, which will adversely affect the analysis results. On the contrary, when the light level is too low, the bacterial culture studied will develop very slowly. This situation will cause time loss. It was decided that the optimum light level in the laboratory to be measured would be 60%-80% by taking advantage of the previous experience of the project team performing the analysis.

Table 5.1 Labels and Limit values to be used for the SITARC dataset

	Excellent (5)	**Good (4)**	**Moderate (3)**	**Poor (2)**	**Terrible (1)**
Temperature	19-21	18-19	17-18	16-17	<16
		21-22	22-23	23-24	>24
Humidity	40-60	30-40	20-30	10-20	<10
		60-70	70-80	80-90	>90
CO_2	<700	700-900	900-1,100	1,100-1,300	>1,300
TVOC	<40	40-70	70-100	100-150	>150
$PM_{2.5}$	<10	10-20	20-30	30-40	>40
PM_{10}	<20	20-40	40-60	60-80	>80
CO	<25	25-50	50-75	75-100	>100
Light	Nan	Nan	Nan	Nan	Nan

The optimum levels of the parameters evaluated within the scope of this study were presented above. In other words, considering these optimum levels, the number of classes planned to be created in the Data Labeling section can be considered as two (For example Good, Poor). However, in order for the proposed system to provide a common framework apart from domains that are used within the scope of this study,

the rows in the data are divided into 5 different classes. Another factor in the separation of rows into so many classes is to push the limits of the models to be tried as a prediction approach. Generated classes and their limit values are shown in Table 5.1.

In many studies in the literature, generally, a few parameters and two different classes are used, such as “Good” and “Poor” (Adeleke et al., 2017) Since the most prominent purpose in this study is to find a suitable estimation algorithm for ontological sensor data, the situation for the algorithms to be selected is made a little more difficult, 5 different classes are defined for 8 parameters and the limit values are determined. The class of an instance is determined by the parameter with the worst class value among the parameters that make up that row. Table 5.2 shows how the class value of the row is determined.

Table 5.2 Determining the class values of parameters and rows

Temp.	Humidity	CO_2	TVOC	$PM_{2.5}$	PM_{10}	CO	Light	Nominal
22.93	54.16	534.55	20.86	10.66	12.85	27	74.63	Moderate
23.01	53.78	541.1	21.68	10.09	11.83	27	67.1	Poor
21.03	42.12	422	2.48	0.88	1.12	21.6	26	Good
20.99	42.2	417.45	1.71	1.32	1.38	21	4	Excellent
20.27	50.94	879.46	71.31	5.08	5.78	32.59	78.07	Moderate
20.31	50.94	554.24	23.08	4.67	5.73	32.8	76.56	Good
20.25	52.34	1,348.59	142.37	7.58	8.96	37.28	28	Terrible
20.31	52.3	1,223.55	128.47	7.79	9.22	34.65	28	Poor
19.66	52.25	1,306.33	138.5	6.53	7.71	255.35	79.43	Terrible
19.59	55.33	407.04	0.28	3.42	3.73	22.57	26	Excellent

In SITARC, when the rows are classified according to the above rules, it has been seen that 65% of the total of 3168 rows of data are at the desired level for the laboratory indoor environments. However, in the remaining 35%, timely preparation of necessary action plans is vital for laboratory analysis results, and employee health.

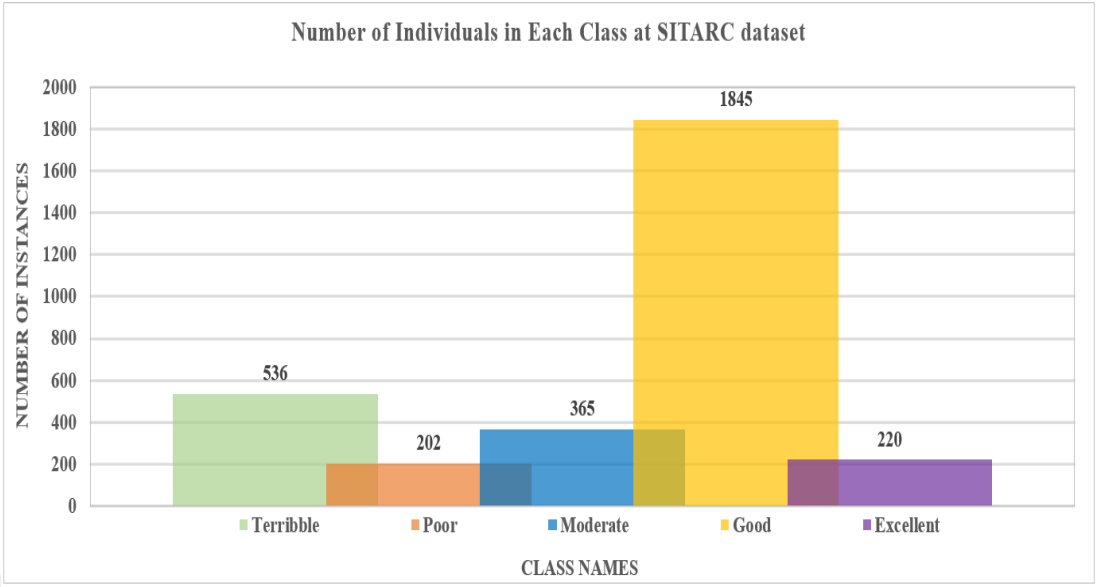


Figure 5.17 Distribution of lines in SITARC dataset to classes

The experiments reveal that the time laboratory air quality is in the desired range when there is no biological analysis and nobody is in the environment. After labeling the rows in the dataset created from the data collected in SITARC, their distribution to the previously defined classes is given in Figure 5.17.

Table 5.3 Labels and Limit values to be used for the MICU dataset

	Excellent (5)	**Good (4)**	**Moderate (3)**	**Poor (2)**	**Terrible (1)**
Temperature	23-25	22-23	21-22	20-21	<20
		25-26	26-27	27-28	>28
Humidity	30-70	25-30	20-25	15-20	<15
		70-75	75-80	80-85	>85
CO_2	<700	700-900	900-1,100	1,100-1,300	>1,300
TVOC	<40	40-70	70-100	100-150	>150
$PM_{2.5}$	<10	10-20	20-30	30-40	>40
PM_{10}	<20	20-40	40-60	60-80	>80
CO	<30	30-50	50-75	75-100	>100
Light	Nan	Nan	Nan	Nan	Nan

Since high-temperature values negatively affect the results of laboratory analysis studies, the temperature value was kept slightly below normal conditions while determining the labels of the measurements made in SITARC. However, during the monitoring and treatment of patients in MICU, the temperature values are increased

by 2 °C compared to normal indoor temperature conditions due to the lack of clothing or thinness. Therefore, while labeling the data in MICU, the values in Table 5.3 were taken as the criterion.

In MICU, when the rows are classified according to the above rules, it has been seen that 40% and of the total of 2780 rows of data are at the desired level for the hospital indoor environments. In addition to this data, approximately 32% of all of the data are within reasonable average values. However, in the remaining, approximately 28%, timely preparation of necessary action plans is vital for hospital staff and especially patients who are in the process of monitoring and treatment there. According to the results of the collected data, it is seen that the environmental conditions at MICU generally worsen in the morning hours that are patient control time, and in the afternoon, hours visited by the relatives of the patients. In the other remaining times, MICU indoor air quality level was seen that generally healthy. After labeling the rows in the dataset created from the data collected in MICU, their distribution to the previously defined classes is given in Figure 5.18.

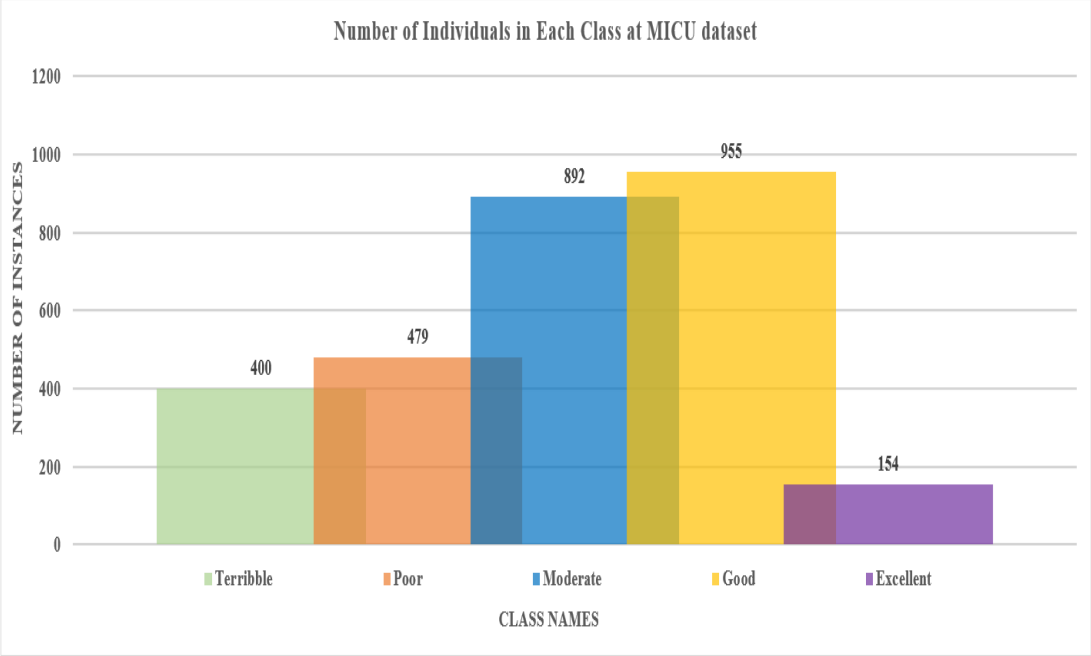


Figure 5.18 Distribution of lines in MICU dataset to classes

5.7.1.3 Normalization

Many models we will use in the prediction phase use absolute distance measurement methods such as Euclidean and Minkowski. Therefore, it may be necessary to apply normalization processes to each feature, especially when using prediction approaches in multi-attribute data sets. A model inevitably applied without normalization will be affected by a high weight attribute even if there is no correlation between them.

The measurement ranges, so limit values, of each of the sensors used in this study are different. The measuring range is the total range that the instrument can measure under normal conditions. Table 5.4 shows the maximum and minimum values that can be measured by the sensors used in this study. Since the ranges of the parameters used for the proposed prediction models are in very different ranges, a normalization approach accepted in the literature must be applied to the dataset before proceeding to the model stage.

Table 5.4 Value ranges of measured parameters

No	Sensor	Parameter	Unit	Measurement Range
1	DHT22	Temperature	°C	-40 °C-125 °C (± 0.5)
2	DHT22	Humidity	% rh	0%-100% (± 2.5-5)
3	CCS-811	Carbon Dioxide	ppm	400-29,206 ppm
4	CCS-811	Total Volatile Organic Compounds	ppb	0-32,768 ppb
5	Nova PM	Particular Matter 2.5	ppm	0.0-999.9 ppm
6	Nova PM	Particular Matter 10	ppm	0.0-999.9 ppm
7	MQ-7	Carbon Monoxide	ppm	10-10,000 ppm
8	LDR	Light Level	%	0%-100%

As can be seen in Table 5.4, the values of some parameters can be between 0 and 100, while some parameter values can go up to 10,000. Therefore, it is certain that the prediction algorithms will decide according to the parameter with large values. In order to prevent this situation and to ensure that the parameters affect the estimation

algorithm equally, all parameters were implemented min-max normalization approaches.

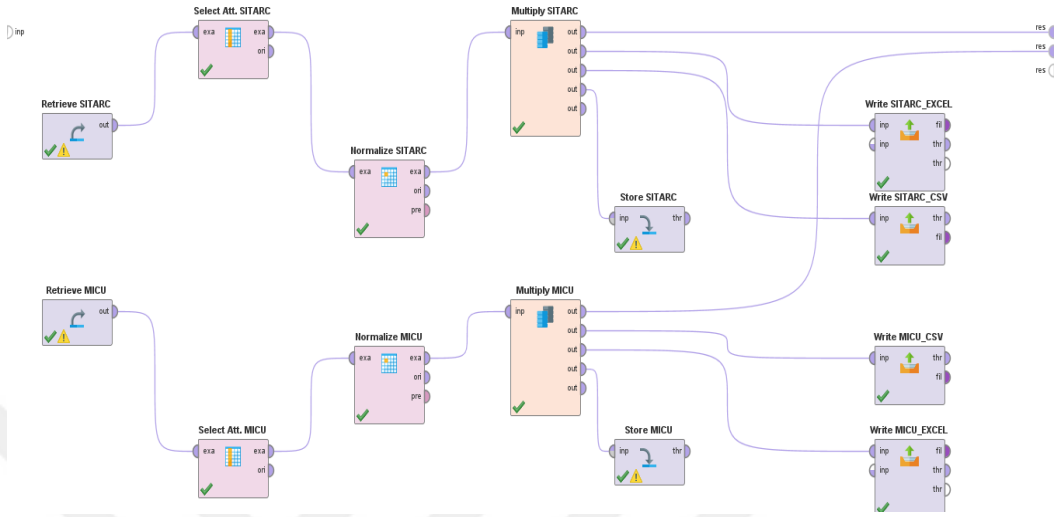


Figure 5.19 The Normalization Process of SITARC and MICU in RapidMiner

The Min-Max method is the most known and used normalization methods. For each attribute included in the dataset, the minimum value of that attribute gets transformed into 0, and its maximum value gets transformed into 1. Finally, all other intermediate values for that feature are shifted to the range [0-1]. The shifting process of parameters in MICU and SITARC data sets to range [0-1] is given in Figure 5.19.

In Figure 5.19 given above, the columns to be normalized in both data set used within the scope of the project were selected by "Select Attribute". During this selection, “DateTime”, “Status_Nominal”, and “Status_Numeric” columns were not selected since they will not be subjected to any normalization process. The component used for the normalization process of Rapid Miner was taken into the development environment and the normalization method was determined as range transformation. The minimum value of the Range transformation method is set to 0 while the maximum value is set to 1.

The new datasets created as a result of normalization have been saved to Local Repositor via the Store component to be used in further operations. The results are also

saved in different formats for different purposes with the WriteExcel and WriteCvs components provided by the Rapid Miner studio.

5.7.1.4 Outlier Detection

An outlier can be defined as an observation that differs from other observations collected in a parameter (column) in the data set. It may be impossible to distinguish precisely which value in a data set is the error and which value is the actual measurement. However, it is certain that it will affect the result negatively in the prediction phase. This Ph.D. Outlier values in the data collected in two media selected within the scope of the thesis study were generally caused by sensor measurement error or communication between nodes.

Sometimes outliers can be caused by human error. For example, in an environment where the light parameter is measured, if someone prevents the sensor from receiving light with a physical object, this is a human error that causes the sensor value to deviate downward. Both system-based and human-induced errors cause the prediction to be biassed and inaccurate. For this reason, analyzing the collected data and eliminating some possible inconsistent parts will increase the prediction success of the model used in the prediction phase.

In the Rapid Miner Studio development environment, there are operators based on different approaches to find outliers in the data. Some of them are Distance-Based Outlier Detection Operator, Density-Based Outlier Detection Operator, Local Outlier Factor (LOF), and Class Outlier Factor (COF). The detection process of the data in datasets created within the scope of the project was carried out in two stages. In the first of these, the parameters are evaluated within themselves and the outliers in that attribute (Column) have been eliminated.

Firstly Attribute-based outlier detection was performed after the normalization process like dataset-based outlier detection. However, it has been noticed that some sensor data measure very extreme values due to sensor measurement errors. It is certain

that this situation will negatively affect the attribute with extreme values during normalization.

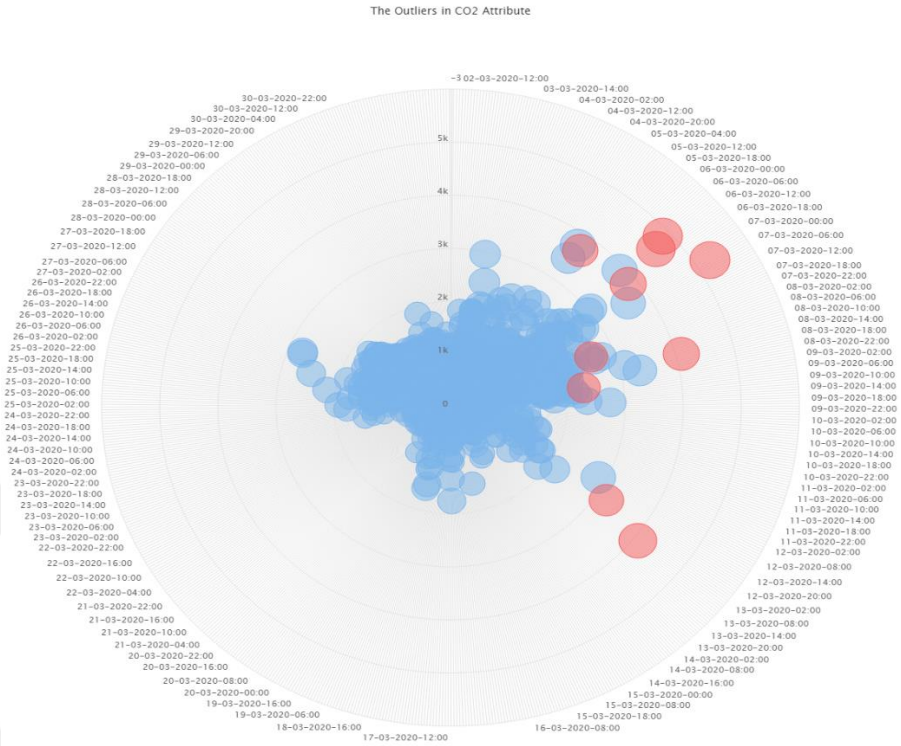


Figure 5.20 Graphical display of Outliers in CO_2 attribute from MICU Dataset

For example, let's assume that the value of the light parameter originating from sensor measurement is accidentally measured as 14,345,987. This observatory value will be the highest value of that parameter in the normalization phase, that is, it will be transformed to 1. Accordingly, the remaining values of the Light parameter, which takes a normal value in the range 0-100, will be proportioned and will probably be transformed to a value that is very close to the value 0 or 0. In Figure 5.20, outliers detected in CO_2, which is one of the parameters that make up the MICU dataset, are given.

For this reason, attribute-based outlier detection was performed at the very beginning of data preprocessing processes in order to prevent the erroneous data from affecting normalization and data labeling. Distance-based approaches were used while outlier detection was performed from the Attribute perspective. Among the distance-based approaches, Euclidian Distance, Cosine Distance, and Squared Distance

algorithms, which are frequently used in the literature, have been tried. Although there is no difference in the result due to the algorithm, the Cosine Distance algorithm has been preferred to eliminate the outliers in the datasets. 10 values that constitute inconsistency for every parameter has been eliminated.

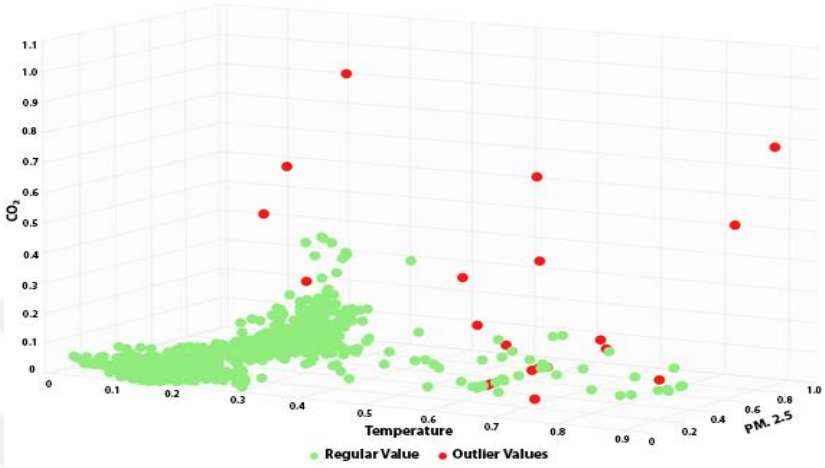


Figure 5.21 Outliers detected on the labels in the SITARC dataset

In the second step of the Outlier detection process, after the class label of each transaction (line) has been assigned, outliers are identified and eliminated through this class label. While determining outliers, the K-NN neighborhood approach was used (k = 10) and a total of 20 observation data were eliminated. The graphic of the outlier detection process performed by including all parameters on the label column in the SITARC data set is given in Figure 5.21.

5.7.2 Model Selection for Prediction

After preprocessing operations are applied to the created datasets, the last step for a proactive system design is to determine the appropriate model for the datasets. There are numerous prediction approaches for different situations and different datasets in the literature. The main reason for the existence of so many prediction approaches and algorithms in the literature is that there is no specific model to guarantee the result for each dataset and use-case. In order to be able to say that an algorithm is the best for a case study or dataset, it must create a sensitive balance especially in terms of accuracy, training time, test duration, flexibility. In other words, it can be quite difficult to predict

which approach will work better for each different situation and each different use-case.

Therefore, many attempts have been made using algorithms frequently used in ML applications while determining the prediction algorithm on datasets created from the data collected within the scope of this thesis. For ontological datasets created as a result of the experiments, the best prediction algorithms were determined and suggested by considering the criteria that show the quality of the algorithm used for that dataset, such as accuracy, performance, flexibility. Data sets containing sensor data generally contain numerical types. Although it may seem easy to understand digital data types, finding patterns can be difficult when it comes to large data heaps. Therefore, sometimes finding the algorithm that can work optimally for a data set and integrating it into the data set becomes more complex than it seems. This section will focus on the experiments performed to determine the prediction algorithm, and the classical methods used.

When implementing ML algorithms into a dataset, kernel function selection, parameter optimization, training data selection, and test data selection are important steps that affect the result. While determining the suitable models for the datasets used, the Auto Model owned by Rapid Miner has been used and although the algorithms and parameters used have been mentioned in detail before, this section is briefly mentioned below in order to better understand this section.

5.7.2.1 Naive Bayes Algorithm

The NB technique takes its name from Thomas Bayes and his conditional probability theorem. It is one of the oldest supervised learning algorithms among ML methods. One of its most important features is its simple operation and speed. The algorithm accepts all variables as independent, but this assumption is seldom valid in the real world. The performance information of the NB algorithm on the SITARC data set is given numerically in Table 5.5.

Table 5.5 Performance of Naive Bayes algorithms on SITARC RDF dataset

Prediction	True Terrible	True Poor	True Moderate	True Good	True Excellent	Class Precision
Prediction Terrible	152	5	3	0	0	95.00%
Prediction Poor	0	48	47	199	37	14.50%
Prediction Moderate	0	3	60	257	14	17.96%
Prediction Good	0	0	0	70	10	87.50%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	100.00%	85.71%	54.55%	13.31%	0.00%	

The NB approach was implemented on the SITARC RDF data set and the MICU RDF data set, respectively. The accuracy performance of this approach on the specified datasets was mediocre values of 36% and 26%, respectively. As seen in Table 5.5 and Table 5.6, the performance of the prediction made by the NB algorithm is very low.

Table 5.6 Performance of Naive Bayes algorithms on MICU RDF dataset

Prediction	True Terrible	True Poor	True Moderate	True Good	True Excellent	Class Precision
Prediction Terrible	110	76	140	84	2	26.70%
Prediction Poor	2	30	33	0	0	46.15%
Prediction Moderate	0	42	77	184	44	22.19%
Prediction Good	0	0	0	0	0	0.00%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	98.21%	20.27%	30.80%	0.00%	0.00%	

While this approach can be considered successful in distinguishing the "Terrible" and "Good" classes, it has performed poorly in distinguishing the other classes. As a result, the use of the NB algorithm in SITARC and MICU ontological datasets developed with semantic technologies is definitely not appropriate. In Table 5.6, the

performance evaluation of the Naive Bayes algorithm on the SITARC data is given numerically.

Some parameters affect all algorithms tested on the dataset more or less than others. In the implementation of the Naive Bayes algorithm on the SITARC data, $PM_{2.5}$ and PM_{10} attributes are the most influential parameter on the result. The parameter that affects the result least is TVOC.

5.7.2.2 Generalized Linear Model (GLM) Algorithm

GLM is a method developed by John Nedler and Robert Wedderburn by combining various statistical models. This model is a flexible and generalized form of ordered linear regression that can classify regardless of the normal distribution of the dependent variable. When the GLM model is applied to the SITARC data set, it has proven its usability for RDF data sets enriched by using ontologies, providing an accuracy rate of 81%. When applied to the GLM model MICU data set, it provided an average performance of 58%.

Table 5.7 Performance of GLM algorithms on SITARC RDF dataset

Prediction	**True Terrible**	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	0	0	1	0	135	99.26%
Prediction Poor	0	2	0	28	2	87.50%
Prediction Moderate	0	84	18	25	3	64.62%
Prediction Good	23	23	451	3	12	88.09%
Prediction Excellent	38	1	56	0	0	40.00%
Class Recall	62.30%	76.36%	85.74%	50.00%	88.82%	

GLM is a regression-based method and it is clear that regression-based methods are particularly effective in terms of uptime. Therefore, one of the most defining features for GLM has been accuracy performance. In Table 5.8, the performance of the GLM

algorithm in the classes on the SITARC data set is given numerically. Looking at Table 5.8, the class that the GLM algorithm is most successful in terms of distinctiveness is a "Terrible" class, while the most unsuccessful class is the "Excellent" class.

Table 5.8 Performance of GLM algorithms on SITARC RDF dataset

Prediction	True Terrible	True Poor	True Moderate	True Good	True Excellent	Class Precision
Prediction Terrible	80	1	0	0	0	98.77%
Prediction Poor	31	80	35	1	0	54.42%
Prediction Moderate	4	29	63	18	0	55.26%
Prediction Good	2	28	149	253	41	53.49%
Prediction Excellent	0	1	1	3	4	44.44%
Class Recall	68.38%	57.55%	25.40%	92.00%	8.89%	

On the other hand, looking at GLM performances in the MICU dataset, it is seen that the most successful class is the "Terrible" class in parallel with the SITARC dataset. Again, as in the SIATRC dataset, the most unsuccessful class distinction is seen as the "Excellent " class in the MICU dataset. In Table 5.8, the performance of the GLM algorithm in the classes on the MICU data set is given numerically and the discrimination ability on the classes is shown.

5.7.2.3 Logistic Regression (Logit) Algorithm

In logistic regression, as in other regression models, the aim is to establish a model with a certain number of variables and with an acceptable error rate. It is preferred for multivariate data, especially if the dependent variable is not continuous. The main difference between linear regression and logistic regression is to estimate the value of the dependent variable in linear regression, while the probability of realization of the values that the dependent variable can take is calculated in the logistic regression. Therefore, logistic regression takes values between 0 and 1. In addition, linear

regression uses the Ordinary Least Squares (OLS) method for estimating, logit uses the Maximum Likelihood (MLE) method.

Table 5.9 Performance of Logit algorithms on SITARC RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	4	0	0	0	0	100.00%
Prediction Poor	1	0	0	0	0	0.00%
Prediction Moderate	37	57	86	84	13	31.05%
Prediction Good	112	1	6	4	0	3.25%
Prediction Excellent	0	0	12	437	51	10.20%
Class Recall	2.60%	0.00%	82.69%	0.76%	79.69%	

After the logit approach is implemented in datasets, it is obvious that logit is the most unsuccessful algorithm when the accuracy performance of both data sets is taken as the mean. Logit algorithm is highly affected by repetitive data. Although hourly averages of sensor measurements are added within the scope of this study, most of the time the measured average values can be very close to each other. For this reason, it is clearly seen in this thesis study that the logit algorithm is not suitable for data sets consisting of time series such as sensor data. In Table 5.9, the accuracy performance measurements of the Logit algorithm on the SITARC dataset are given numerically from the perspective of classes.

In Table 5.9, it is seen that the capacity of the Logit algorithm to distinguish almost all classes in the SITARC dataset is low. Although accuracy performance seems to be perfect for the "Terrible" class, only 4 lines (individual) had been selected as members of this class in the randomly selected test algorithm. The 4 lines are insufficient to comment on the performance of a ML algorithm in the class perspective. In Table 5.10, the accuracy performance measurements of the Logit algorithm on the MICU dataset are given numerically from the perspective of classes.

Table 5.10 Performance of Logit algorithms on MICU RDF dataset

	True Terrible	True Poor	True Moderate	True Good	True Excellent	Class Precision
Prediction Terrible	29	0	0	0	0	100.00%
Prediction Poor	62	37	5	0	0	35.58%
Prediction Moderate	4	76	186	152	6	43.87%
Prediction Good	11	32	61	129	34	48.31%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	27.36%	25.52%	73.81%	45.91%	0.00%	

In Table 5.10, it is seen that the capacity of the Logit algorithm to distinguish almost all classes in the MICU dataset is low, as in the SITARC dataset. Considering the row number in the Terrible class together with the row number of the "Terrible" class in the SITARC dataset, it is seen that the logit algorithm has a high distinctiveness over a single Terrible class on these classes.

5.7.2.4 Fast Large Margin (FLM) Algorithm

Algorithms such as Support Vector Machines that position the decision border in order to maximize the distance between two classes are called Large Margin algorithms. In other words, data estimated to belong to separate classes are mapped to have as clear a distance as possible. This type of linear classifiers can easily work with multidimensional data sets. The larger the natural margin between classes, the higher the success of the classifier.

As it is understood, the Fast-Large Margin approach is a specialized algorithm belonging to the Support Vector Machine family. The average results obtained in the implementation of FLM into two ontological sensor data sets created within the scope of this thesis study are thought to be better than the average results obtained from many other algorithms. While the accuracy performance of FLM on the SITARC data set is 74%, the accuracy performance on the MICU data set is 63%. Although the FLM

algorithm failed to be the best algorithm separately for both data sets, it succeeded to be the 2nd best algorithm among other applied algorithms in terms of accuracy performance on average. Table 5.11 gives the numerical value of the accuracy performance of the FLM algorithm on the SITARC data set from the perspective of class labels.

Table 5.11 Performance of FLM algorithms on SITARC RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	141	0	0	0	0	100.00%
Prediction Poor	0	0	0	0	0	0.00%
Prediction Moderate	0	0	0	0	0	0.00%
Prediction Good	14	55	107	524	55	69.40%
Prediction Excellent	0	0	0	0	9	100.00%
Class Recall	90.97%	0.00%	0.00%	100.00%	14.06%	

In Table 5.11, when the performance of the implementation of the FLM algorithm on the SITARC data set is evaluated separately on the class labels, the class distinction power can be evaluated as very good for "Terrible" and "Excellent" classes, and above average for the "Good" class. Class discrimination performance in the SITARC dataset of this approach is not possible to evaluate as there is no row to predict for these classes due to random selection of test and training set for Good and Poor classes. However, the discriminative power of the FLM algorithm for these classes was evaluated in the MICU dataset.

In Table 5.12, the numerical value of the accuracy performance of the FLM algorithm on the MICU data set is given from the perspective of class labels. Considering the performance of implementing the FLM algorithm into the MICU data set on class labels separately, the class discrimination power can be evaluated as very good for the "Terrible" class, not measurable for the" Excellent" class, and above average for the other classes.

Table 5.12 Performance of FLM algorithms on MICU RDF dataset

	True Terrible	True Poor	True Moderate	True Good	True Excellent	Class Precision
Prediction Terrible	87	3	0	0	0	96.67%
Prediction Poor	18	32	1	0	0	62.75%
Prediction Moderate	12	84	148	25	0	55.02%
Prediction Good	0	20	99	250	45	60.39%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	74.36%	23.02%	59.68%	90.91%	0.00%	

In these experiments using the auto model of Rapid Miner Studio data processing and ML application, the default parameters applied by many algorithms while using the auto model were not changed. However, to improve the results, algorithms working on parameters such as parameter SVM and Random Forest were tested with other parameters and new comparisons were created. Another algorithm in which the parameter is effective is FLM. The C cost parameter used by this approach in prediction calculations was evaluated using different values.

The default C parameter that Rapid Miner Studio uses for the SITARC data set is 0.001. In Figure 5.22, Error Ratings of different values tried for optimum C parameter for SITARC data set are given. However, as can be seen from Figure 5.22, it is seen that the best C value for the FLM algorithm on the SITARC dataset is 10 and 100. In Figure 5.23, Error Ratings of different values tested for optimum C parameter on MICU data are given. When these results are taken as reference, it is seen that the best C value that can be used for the MICU data of the FLM algorithm is 10.

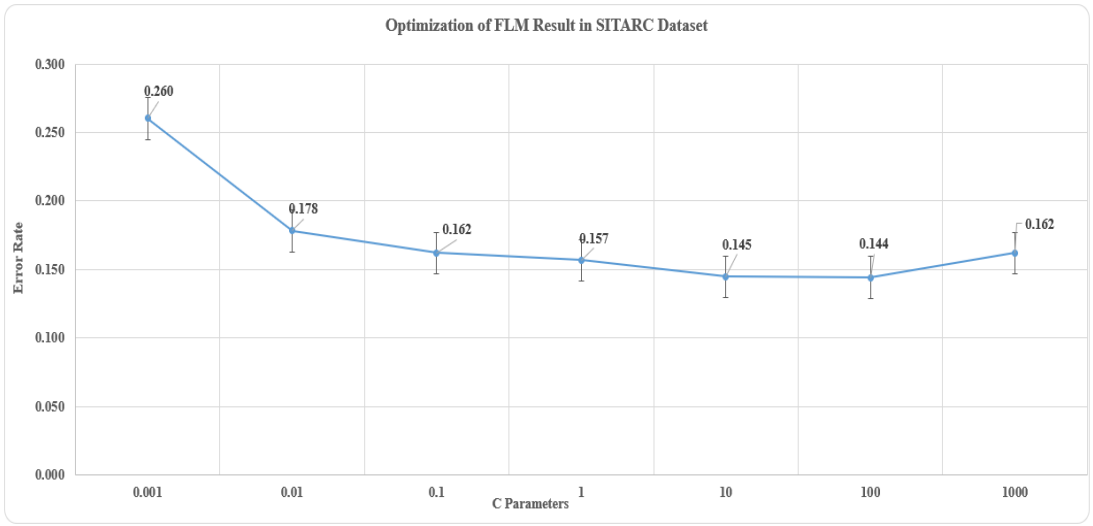


Figure 5.22 The results against different C values in the SITARC dataset

The success of the FLM algorithm certainly increases, even more, when the optimization is made other than the C parameter assigned by Rapid Miner Studio in the auto model. As a result of operating the C parameter at the optimum level in both datasets, the accuracy performance of the FLM algorithm has increased to 85% in the SITARC dataset and 65% in the MICU dataset. Although the accuracy rates increase after the optimization of the FLM algorithm, it remains in second place in terms of accuracy performance when the averages in both data are considered and compared to the averages of other algorithms.

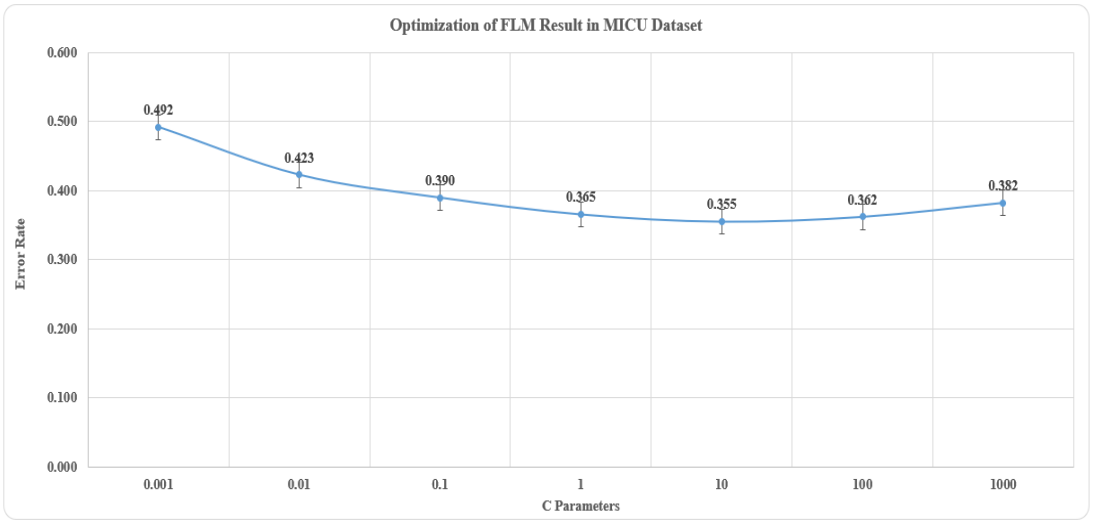


Figure 5.23 The results against different C values in the MICU dataset

5.7.2.5 Deep Learning (DL) Algorithm

Deep Learning is an emerging ML technique that has become popular recently. DL is closely related to the artificial neural network (ANN). ANN is the general name of the algorithms that learn to generalize the whole data set from a small data set by modeling the working principles of the human nervous system and the brain. It contains mechanisms, like humans that ensure making decisions about situations that they have not seen based on their past experiences. In other words, ANN is used to model the relationship between input data and output data. DL is a more sophisticated and structurally more complex form of ANN. Because DL has more intermediate layers, learning is relatively slow compared to ANN and requires more processing power.

Within the scope of this thesis study of DL, it is seen that the average results obtained in the implementation of two semantically enriched sensor data sets created using semantic technologies are better than the average results obtained from other algorithms. While the accuracy performance of DL on the SITARC data set is 89%, the accuracy performance on the MICU data set is 68%.

Table 5.13 Performance of DL algorithms on SITARC RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	149	2	0	0	0	98.68%
Prediction Poor	2	42	8	0	0	80.77%
Prediction Moderate	0	13	87	19	0	73.11%
Prediction Good	0	0	9	486	23	93.82%
Prediction Excellent	0	0	0	24	41	63.08%
Class Recall	98.68%	73.68%	83.65%	91.87%	64.06%	

It is the algorithm that provides the best accuracy performance among all algorithms in the DL MICU data set. Although it is not the best performing algorithm in the

SITARC dataset, it shows approximate accuracy performance value with the Random Forest algorithm that provides the best performance. Table 5.13 shows the numerical value of the accuracy performance of the DL algorithm on the SITARC data set from the perspective of class labels.

When the data in Table 5.13 is taken as reference, the classes in which the DL algorithm has the best discrimination power are "Terrible" and "Good" classes, while for the "Poor" and "Moderate" classes, it has provided a much higher accuracy performance. Although the class determining power of the DL algorithm is the class with the weakest "Excellent" class, its accuracy performance is still above average.

Table 5.14 Performance of DL algorithms on MICU RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	82	0	0	0	0	100.00%
Prediction Poor	29	95	19	0	0	66.43%
Prediction Moderate	1	52	210	96	0	58.50%
Prediction Good	0	1	21	172	44	72.27%
Prediction Excellent	0	0	0	0	2	100.00%
Class Recall	73.21%	64.19%	84.00%	64.18%	4.35%	

In Table 5.14, the numerical value of the accuracy performance of the DL algorithm on the MICU data set is given from the perspective of class labels. From both the class separation table in the SITARC data set and the class separation table in the MICU dataset, it shows that the DL algorithm is an algorithm that might be used on ontological sensor data.

Considering the data in Table 5.14, it is certain that the DL algorithm has a high performance in distinguishing the "Terrible" class for the MICU data set. It achieved higher than average success in distinguishing "Poor" and "Good" classes and close to average in distinguishing "Moderate" class. There is an insufficient number of rows in

the test class to comment on the distinguishing feature of the DL algorithm for the "Excellent" class.

5.7.2.6 Decision Tree (DT) Algorithm

DT is one of the most established algorithms of ML and data mining. It is used in both classification and regression analysis. The most important feature of DT is that it simplifies and clarifies the decision-making mechanism in any process. It handles the decision-making process like a tree structure and its name comes from also this characteristic. It recursively divides the search space into subsets according to an attribute in each decision node. The division process ends when the data remaining in the subset cannot be separated according to any attributes. The lowest node of the tree specifies the classes. To establish an optimal DT is often an NP-Complete problem. Therefore, it requires applying heuristic ways to establish a good near-optimal DT.

The average results obtained in the implementation of DT into two ontological sensor data sets created within the scope of this thesis study were compared separately with the averages of the results obtained from other algorithms. As a result of this comparison, while the accuracy performance of DT in the first dataset, SITARC, was above the average, it was below the average in the other ontological data set, MICU.

Table 5.15 Performance of DT algorithms on SITARC RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	152	1	0	0	0	99.35%
Prediction Poor	1	49	6	0	0	87.50%
Prediction Moderate	0	4	84	21	0	77.06%
Prediction Good	0	0	19	509	59	86.71%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	99.35%	90.74%	77.06%	96.04%	0.00%	

Although the parameters in the datasets used and the algorithm applied are the same, the fact that they have completely different results is due to the character of the datasets. Table 5.15 shows the numerical value of the accuracy performance of the DT algorithm on the SITARC data set from the perspective of class labels.

While the accuracy performance of the DT algorithm on the SITARC data set is 88%, the accuracy performance on the MICU data set is 43%. According to this Table 5.15, it can be said that the tribe of distinguishing all other classes except the "Excellent" class in the SITARC data set of the DT algorithm is high. For the "Excellent" class, it is impossible to comment on the aforementioned reasons. Table 5.16 gives the numerical value of the accuracy performance of the DT algorithm on the MICU data set from the perspective of class labels.

While the DT algorithm is implemented with Rapid Miner Studio auto module, "Maximal Depth" is given as a default of 20. However, giving "Maximal Depth" as 20 increases the training and scoring times of Tree algorithms especially. Therefore, for the DT algorithm to compete with other algorithms in terms of time, the value of the Maximal Depth parameter was set to 10 by sacrificing some accuracy. Thus, the time spent by the algorithm in the training and scoring phases has decreased significantly.

Table 5.16 Performance of DT algorithms on MICU RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	152	1	0	0	0	99.35%
Prediction Poor	1	49	6	0	0	87.50%
Prediction Moderate	0	4	84	21	0	77.06%
Prediction Good	0	0	19	509	59	86.71%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	99.35%	90.74%	77.06%	96.04%	0.00%	

DT approach is similar to tree-view flowcharts and decision-making processes are not very complex. In other words, it can be shown schematically how he arrived at the result. For this reason, it is much easier to read and interpret than other data mining algorithms. DT algorithms are widely used because of their easy interpretation, understandability, fast operation, and high reliability of these algorithms. This basic decision support tree, which is formed by interpreting the training data set by the algorithm, is used in making decisions at the scoring stage.

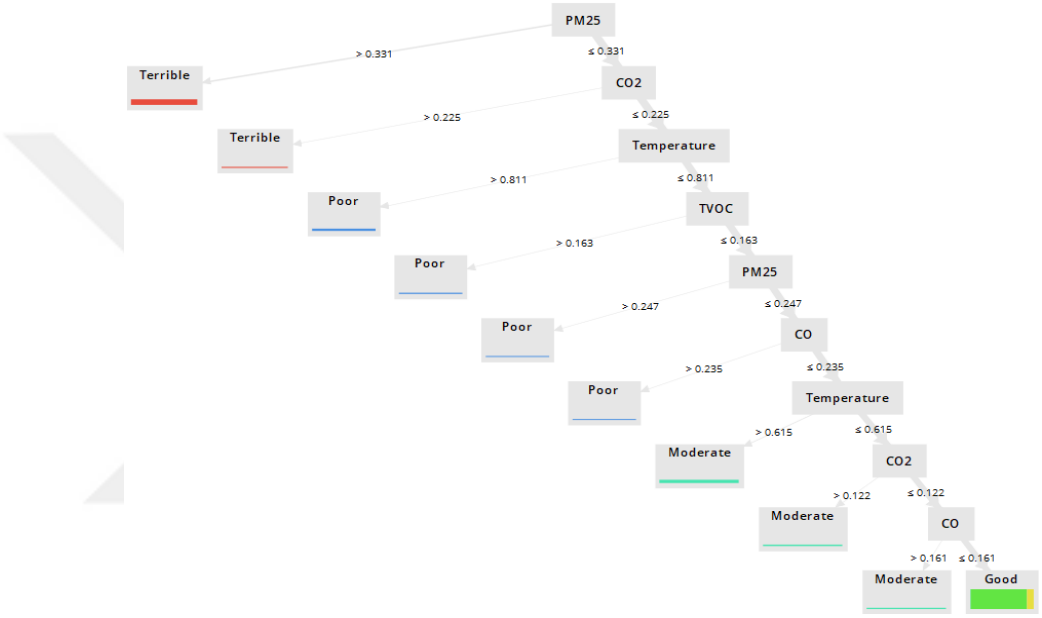


Figure 5.24 The tree model of the DT algorithm for the SITARC training dataset

Therefore, the homogeneous distribution of randomly selected lines in the training data set is vital for the algorithm mechanism to work properly. In Figure 5.24, the tree model obtained by the DT algorithm using the training data set from the SITARC data set and used to make decisions in the test data set is given.

As seen in Figure 5.24, all DTs consists of the trio of root, branch, and leaf. Each of these is called a knot. The main problem for decision tree algorithms is the determination of these nodes. The top node is called the root node. The first problem in creating the DT template is which attribute will be the root node. In general, the attribute that has the most impact on the label should be determined as the root node. Subsequent nodes are also created by considering their effects on the label. The

decision-maker obtained in Figure 5.24 shows a linear flow. But it doesn't mean that DT will be linear for every dataset.

According to Figure 5.24, the root node of this DT has become $PM_{2.5}$. Looking at this root node, when the $PM_{2.5}$ value in the data set exceeds 0.331, the tendency of the corresponding row to have a "Terrible" label has increased. In other words, $PM_{2.5}$ has a great effect on labeling any line as "Terrible". When moving down in DT, at the second node, it is seen that CO_2 has an effect on the label. As the 3rd node, it is obvious that the Temperature attribute has an effect on the label. The rule in this node can be expressed as follows; If the temperature value is greater than 0.811, it can be said that the row generally has a "Poor" label. In Figure 5.25, the tree model that the DT algorithm has obtained using the training data obtained from the MICU data set and used to make decisions in the test data is given.

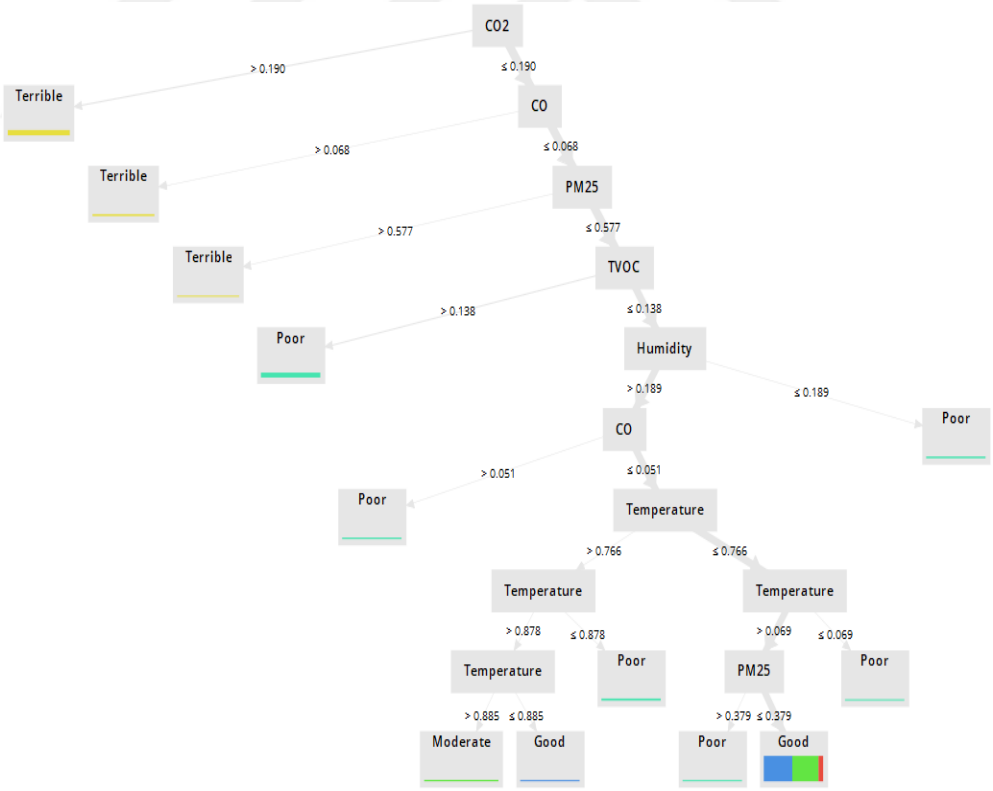


Figure 5.25 The Tree model of the DT algorithm for the MICU training dataset

According to Figure 5.25, the root node of this DT has been CO_2. Looking at this root node, when the CO_2 value in the data set exceeds 0.190, the tendency of the relevant row to have a "Terrible" label has increased. In other words, the CO_2 attribute has a big impact on labeling any line as "Terrible". When moving down in DT, at the second node, it is seen that the CO attribute affects the label. The rule in this node can be expressed as follows; If the value of the temperature attribute is greater than 0.68, it can be said that the line generally has the "Terrible" label. As the 3rd node, it is seen that the $PM_{2.5}$ attribute affects the label. Unlike the DT template obtained from SITARC, the DT template obtained from MICU was not branched linearly after the 5th node.

5.7.2.7 Random Forest (RF) Algorithm

RF is an ensemble learning algorithm that generates multiple decision trees and combines the results obtained from these decision trees with the bagging method. It is one of the popular algorithms used recently because RF can be applied to both regression and classification problems and also achieves good results in these areas. Also, since the RF algorithm trains on different data sets for each feature, overfitting is reduced. The RF algorithm is preferred because it can find the power of distinguishing classes for each feature.

The average results obtained in the implementation of RF into two sensor data sets enriched with semantic web technologies created within the scope of this thesis study were compared separately with the averages of the results obtained from other algorithms. As a result of this comparison, the accuracy performance of RF in the first data set, SITARC, achieved the best performance compared to other algorithms, while it was far below the average in the other ontological data set, MICU.

Just like the results obtained from the implementation of two datasets with DT, in this case where the same parameters are implemented and the same approach is implemented, although the parameters in the two datasets are the same, the data clearly show that the characteristics of the two datasets are very different from each other.

Table 5.17 Performance of RF algorithms on SITARC RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	152	1	0	0	0	99.35%
Prediction Poor	1	52	1	0	0	96.30%
Prediction Moderate	0	1	99	17	0	84.62%
Prediction Good	0	0	9	513	59	88.30%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	99.35%	96.30%	90.83%	96.79%	0.00%	

While the accuracy performance of the RF algorithm on the SITARC data set was 90%, the accuracy performance on the MICU data set was recorded as 26%. Table 5.17 gives the numerical value of the accuracy performance of the RF algorithm on the SITARC data set from the perspective of class labels. According to the data in this table, the ability of the RF algorithm to distinguish all other classes except the "Excellent" class in the SITARC data set can be said to be quite high. It is impossible to comment on the "Excellent" class due to the aforementioned reasons.

Table 5.18 gives the numerical value of the accuracy performance of the RF algorithm on the MICU data set from the perspective of class labels. Considering the data in Table 5.18, it can be said that the RF approach has a good discrimination power of a single "Terrible" class. On the other hand, its power to distinguish between "Poor" and "Moderate" classes is very low. In terms of class discrimination performance in the MICU dataset of this approach, it is not possible to evaluate as there is no row for the "Good" and "Excellent" classes to be predicted, due to the random selection of the test data set as can be seen in Table 5.18.

Table 5.18 Performance of RF algorithms on MICU RDF dataset

	True Terrible	True Poor	True Moderate	True Good	True Excellent	Class Precision
Prediction Terrible	68	4	0	0	0	94.44%
Prediction Poor	49	130	233	208	13	20.54%
Prediction Moderate	0	5	15	67	32	12.61%
Prediction Good	0	0	0	0	0	0.00%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	58.12%	93.53%	6.05%	0.00%	0.00%	

The RF algorithm is also in the Tree algorithms family, and the "Maximal Depth" and "Number of Tree" parameters must be given before the algorithm is implemented. These parameters set the limits when the RF algorithm mechanism attempts to create a model. The given parameters are of great importance in terms of accuracy, training time, and scoring time of the RF algorithm. While the RF algorithm is implemented with Rapid Miner Studio auto module, the "Maximal Depth" and "Number of Trees" values are given as 20 by default.

However, giving the values of the" Maximal Depth" and "Number of Trees" as 20 increases the training and scoring times of the RF algorithm excessively. Therefore, in order for the RF algorithm to compete with other algorithms in terms of time, the value of the "Maximal Depth" and Number of Trees" parameter is set to 10 by sacrificing little accuracy. Thus, the time spent by the algorithm in the training and scoring phases has been significantly reduced. In Figure 5.26, the tree model obtained by the RF algorithm using the training data set from the SITARC data set and used to make decisions in the test data set is given.

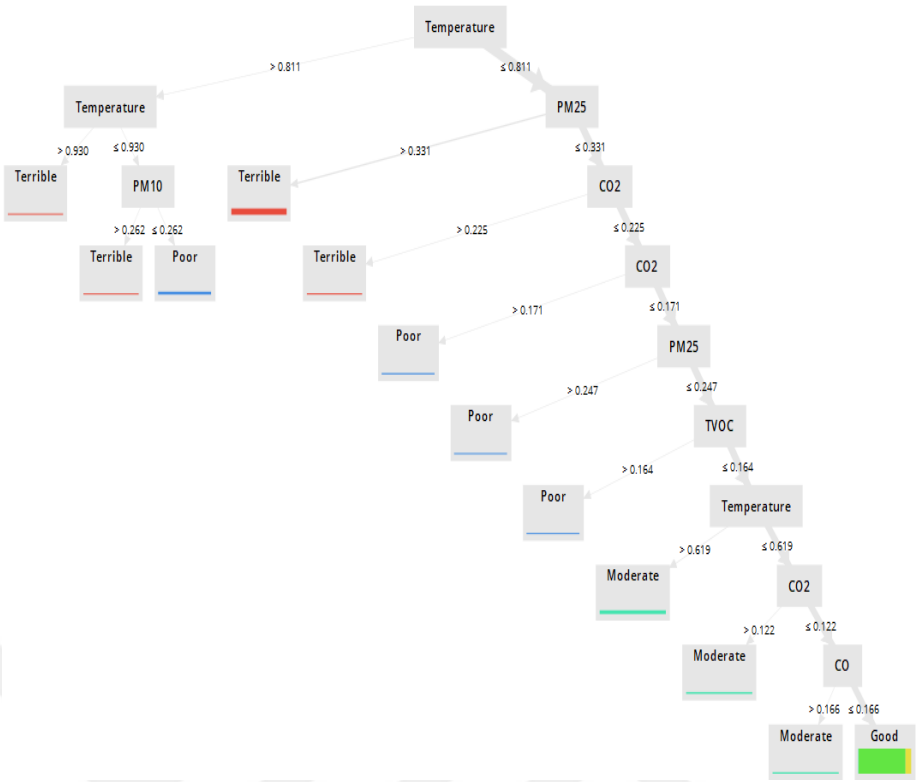


Figure 5.26 The Tree model of the RF algorithm for the SITARC dataset

As seen in Figure 5.26, the root node of this tree structure was the Temperature attribute. Looking at this root node, when the Temperature value in the data set exceeds 0.811, the tree branches to the left, and when it is equal to or below this value, the tree structure is branched towards the right side. It is seen that the second node $PM_{2.5}$ attribute on the right has an effect on the label. The rule in this node can be expressed as follows; If the value of the $PM_{2.5}$ attribute is greater than 0.331, that row can generally be said to have the "Terrible" label. In Figure 5.27, the tree model obtained by the RF algorithm using the training data obtained from the MICU data set and used to make decisions in the test data set is given.

In the tree model created by the RF algorithm shown in Figure 5.27 using the test data set in the MICU data set, the root node was CO_2. Looking at this root node, when the value of the CO_2 attribute in the dataset exceeds 0.190, the tendency of the corresponding row to have a "Terrible" label has increased. In other words, the CO_2 attribute has a big impact on labeling any line as "Terrible".

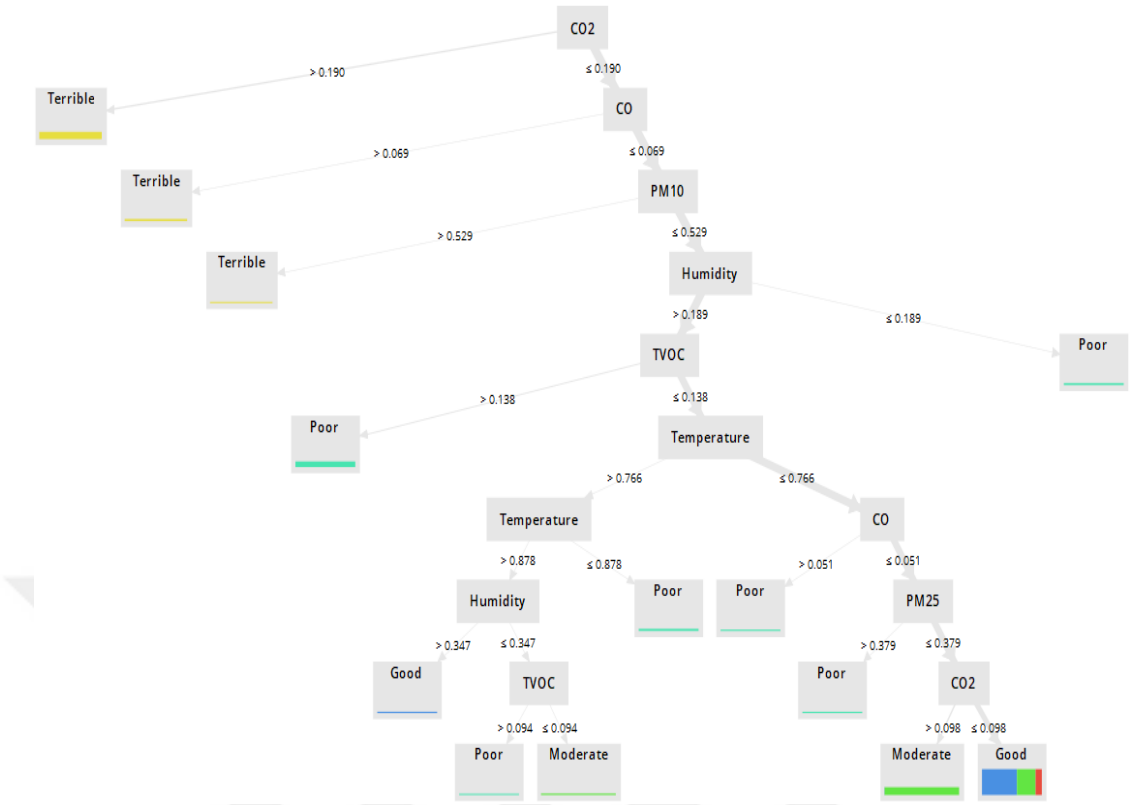


Figure 5.27 The Tree model of the RF algorithm for the MICU dataset

When the RF algorithm moves down the tree model, it is seen that at the second node, the CO is effective on the label. The rule in this node can be expressed as follows; If the CO attribute value is greater than 0.069, it can be said to have a "Terrible" label in general, just like the previous node. As the node, it is obvious that the PM_{10} attribute has an effect on the label. The rule in this node can be expressed as follows; If the PM_{10} attribute value is greater than 0.8529, it can be said that the line generally has a "Terrible" label.

5.7.2.8 Gradient Boosted Trees (GBT) Algorithm

Boosting is a technique used to strengthen weak classifiers. Gradient Boosted Trees, like other techniques described earlier, is used both in regression analysis and classification but is also an ensemble technique that uses decision trees. As the name suggests, the ensemble technique uses the boosting approach, that is, it makes the classification sequential rather than independent. Therefore, this technique tries to

make better predictions using the mistakes of previous estimators. If the terminate criterion is not selected properly, overfitting may be caused, unlike the ensemble bagging technique.

The average results obtained in the implementation of GBT into two ontological sensor data sets created within the scope of this thesis study were compared separately with the averages of the results obtained from other algorithms. As a result of this comparison, while the accuracy performance of GBT in the first data set, SITARC was above average, it was below the average in the other ontological data set, MICU.

While the accuracy performance of the GBT algorithm on the SITARC data set is 80%, the accuracy performance on the MICU data set is 34%. Table 5.19 shows the numerical value of the accuracy performance of the GBT algorithm on the SITARC data set from the perspective of class labels.

Table 5.19 Performance of GBT algorithms on SITARC RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	147	0	0	0	0	100.00%
Prediction Poor	5	55	22	2	0	65.48%
Prediction Moderate	0	1	87	53	0	61.70%
Prediction Good	0	0	1	418	42	90.67%
Prediction Excellent	0	0	0	53	19	26.39%
Class Recall	96.71%	98.21%	79.09%	79.47%	31.15%	

According to data in Table 5.19, the power of the GBT algorithm to distinguish "Terrible" and "Good" classes in the SITARC data set is quite good. The power of distinguishing "Poor" and "Moderate" classes is above average. Considering the results of implementing the GBT algorithm into the SITARC data, it is seen that the ability to distinguish the "Excellent" class is very poor. Table 5.20 shows the numerical

value of the accuracy performance of the GBT algorithm on the MICU data set from the perspective of class labels.

Table 5.20 Performance of GBT algorithms on MICU RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	70	3	0	0	0	95.89%
Prediction Poor	47	131	172	113	9	27.75%
Prediction Moderate	0	5	76	161	36	27.34%
Prediction Good	0	0	0	1	0	100.00%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	59.83%	94.24%	30.65%	0.36%	0.00%	

According to the data presented in Table 5.20, the GBT algorithm has a high ability to distinguish only the "Terrible" class in the MICU data set. The discrimination power of the "Poor" and "Moderate" classes was well below the average. Although the discriminating power of the "Good" class seems very good, there are not enough rows in the test data set for this class to comment on the power of its algorithm to distinguish this class. In terms of class discrimination performance in the MICU dataset of the GBT approach, it is not possible to make an evaluation, as there is no row for this algorithm to be predicted for the "Excellent" class, as can be seen in Table 5.20, due to the random selection of the training set.

The GBT algorithm is in the tree algorithm family, just like DT, and the Learning Rate parameters as well as the Maximal Depth and Number of Tree parameters must be given before the algorithm is implemented. These parameters, like in the RF and DT algorithms, set boundaries when the GBT algorithm tries to create a model. The given parameters are of great importance in terms of accuracy, training time, and test time of the RF algorithm. While the RF algorithm is implemented with Rapid Miner Studio auto module, the maximal depth and Number of Trees values are given as 20 by default. The Learning Rate parameter is given as 0.01.

However, giving the values of "Maximal Depth" and "Number of Trees" as 20 increases the training and test times of Tree algorithms excessively. Therefore, for the GBT algorithm to compete with other algorithms in terms of time, the values of the "Maximal Depth" and "Number of Tree" parameters were set to 10 by sacrificing little accuracy like RF and DT algorithms.

The value of the "Learning Rate" parameter is left as 0.01, which is the default value in Rapid Miner Studio Auto Model. Thus, the time spent by the algorithm in the training and testing phases has been significantly reduced. In Figure 5.28, the tree model obtained by the GBT algorithm using the training data set from the SITARC data set and used to make decisions in the test data set is given.

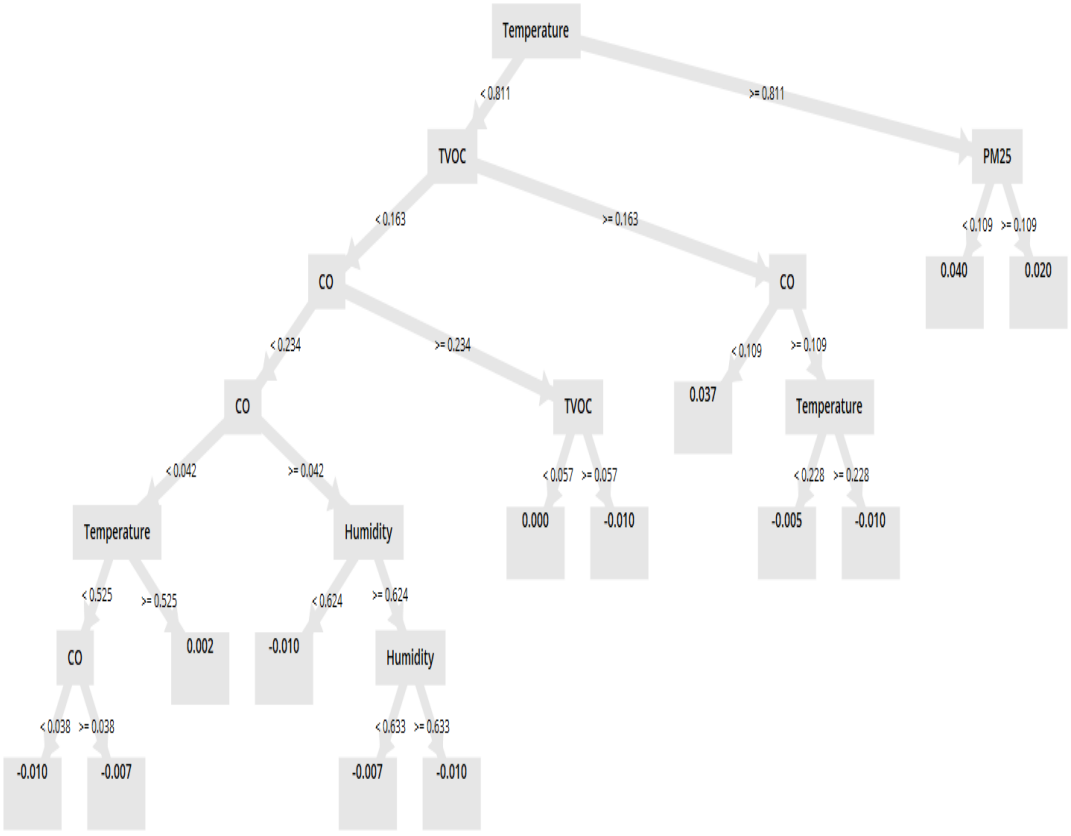


Figure 5.28 The Tree model of the GBT algorithm for the SITARC dataset

As shown in Figure 5.28, the root node of this tree structure has been the Temperature attribute. Looking at this root node, when the Temperature value in the

data set equals 0.811 or passes, the tree branches on the right side, while below this value the tree structure is branched towards the left side. In the second node on the right, it is seen that the $PM_{2.5}$ attribute is effective on the label. The rule in this node can be expressed as follows; If the value of the $PM_{2.5}$ attribute is equal to 0.109 or greater, it can be said that the tree branches to the left side if it is less than 0.109 while the tree branches back to the right. In Figure 5.29, the tree model obtained by the GBT algorithm using the training data obtained from the MICU data set and used to make decisions in the test data set is given.

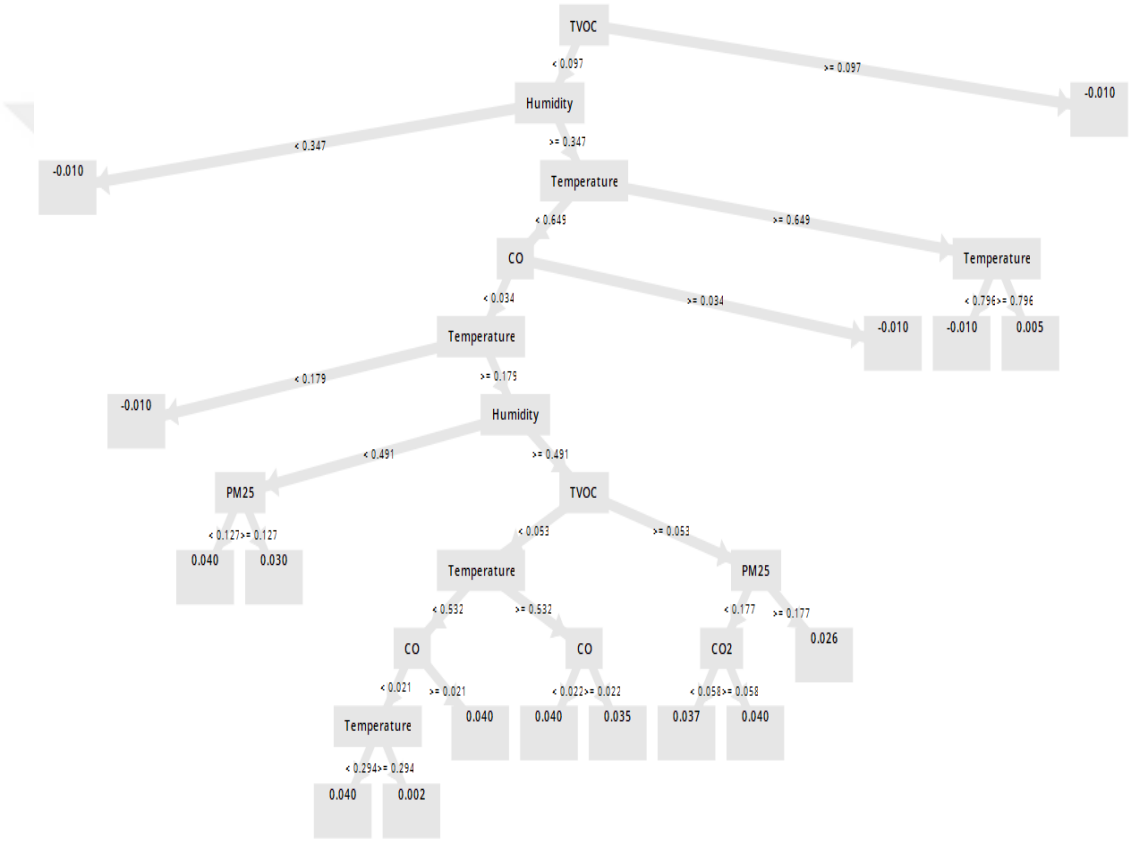


Figure 5.29 The Tree model of the GBT algorithm for the MICU dataset

As seen in Figure 5.29, the root node of this tree structure has been the TVOC attribute. Looking at this root node, when the TVOC attribute value in the data set is equal to or exceeds 0.097, the tree branches to the right, while the tree structure is branched to the left when it is below this value. In the second node on the left, it is seen that the Humidity attribute is effective on the label. The rule in this node can be expressed as follows; If the value of the humidity attribute is equal to or greater than

0.347, it can be said that the tree branches to the left side if it is less than 0.347 while the tree branches back to the right. In the third node on the left, it is seen that the Temperature attribute is effective on the label. The rule in this node can be expressed as follows; If the value of the temperature attribute is equal to or greater than 0.649, it can be said that the tree branches to the left side if it is less than 0.649 while the tree branches back to the right.

5.7.2.9 Support Vector Machine (SVM) Algorithm

In general, it works basically with a similar logic with Logistic Regression. Both approaches focus on finding the best line that separates classes in a data set. It is a nonparametric classifier that takes no parameters. The algorithm allows the line to be drawn to be adjusted in two classes so that it passes from the furthest place to its elements. SVM can also classify linear and nonlinear data but generally tries to classify the data linearly.

The average results obtained in the implementation of SVM into two sensor data sets enriched with semantic web technologies created within the scope of this thesis study were compared separately with the averages of the results obtained from other algorithms. As a result of this comparison, the accuracy performance of SVM in the first dataset, SITARC, was above the average performance when compared to other algorithms, while it was below the average in the other ontological data set, MICU.

Just like the results obtained from the implementation of two data sets such as some algorithms (GBT, RF, DT, etc.) that have been tried and described before, in this case where the same parameters are implemented and the same approach is implemented, the difference is that although the parameters in the two data are the same considering the results, the data clearly show that the characteristics of the two datasets are very different from each other. This situation can be presented as evidence of the unpredictable hypothesis of which algorithm will be best for each case and dataset.

While the accuracy performance of the SVM algorithm on the SITARC data set was 83%, the accuracy performance on the MICU data set was recorded as 45%. The numerical value of the accuracy performance of the SVM algorithm on the SITARC data set from the perspective of class labels is given in Table 5.21.

According to the data in this table, while the discrimination power of the SVM algorithm among the labels that make up the SITARC dataset is very well for the "Terrible" and "Good" classes, the discrimination power on the "Poor" and "Moderate" classes is well above the average. The power to distinguish the SVM approach from the "Excellent" class on this ontological sensor data is well below the average. However, considering the results obtained in this data set, in this dataset, it can be said that the SVM algorithm is an approach that can be used for proactive system design as a prediction model.

Table 5.21 Performance of SVM algorithms on SITARC RDF dataset

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	149	1	0	0	0	99.33%
Prediction Poor	0	55	10	0	0	84.62%
Prediction Moderate	0	5	88	16	0	80.73%
Prediction Good	0	0	3	423	30	92.76%
Prediction Excellent	0	0	0	90	35	28.00%
Class Recall	100.00%	90.16%	87.13%	79.96%	53.85%	

In Table 5.22, the numerical value of the accuracy performance of the GBT algorithm on the MICU data set is given from the perspective of class labels. According to the data presented in Table 5.22, the SVM algorithm has a high ability to distinguish only the "Terrible" class in the MICU data set. While the power to distinguish the "Good" class was slightly above average, the power to distinguish between "Poor" and "Moderate" classes was below average. In terms of the class discrimination performance of the SVM approach in the MICU dataset, it is not

possible to evaluate as there is no row for this class to be predicted, as can be seen in Table 5.22 due to the random selection of the test and training set for the “Excellent” class.

Table 5.22 Performance of SVM algorithms on MICU RDF dataset.

	True Terrible	**True Poor**	**True Moderate**	**True Good**	**True Excellent**	**Class Precision**
Prediction Terrible	87	6	0	0	0	93.55%
Prediction Poor	24	117	174	58	0	31.37%
Prediction Moderate	0	13	72	137	0	32.43%
Prediction Good	0	0	4	93	39	68.38%
Prediction Excellent	0	0	0	0	0	0.00%
Class Recall	78.38%	86.03%	28.80%	32.29%	0.00%	

CHAPTER SIX
EXPERIMENTAL RESULT

6.1 Overview of This Section

In the previous steps of the proposed Ph.D. thesis, two measurement environments were selected and 8 different parameters were collected. Then, these data were enriched with semantic web technologies, and two different RDF datasets were created. Finally, in order to investigate which ML algorithms, have the potential to be used for a proactive system design in RDF datasets, classical ML approaches were tried in these two different RDF datasets, and the results of experiments were shared with the academic community.

In this section, these classical ML algorithms tested on two different RDF datasets will generally be compared, and considering the results, it will be decided which algorithms will be more suitable for which dataset in terms of accuracy, flexibility, and time. In addition, RDF datasets created with data collected from completely different environments will be combined and classical ML algorithms will be implemented on this combined dataset. At the end of this section, the performance of the algorithms implemented on the combined dataset will be discussed in many aspects and a conclusion will be drawn. With this result, it is aimed to shed light on the academic community about classical ML approaches applied to RDF datasets.

6.2 Performance of Classical ML Algorithms on the SITARC Dataset

When the SITARC RDF dataset enriched with semantic web technologies is considered alone, the accuracy performances of the approaches that can be used for prediction in this dataset may be possible to summarize as follows; In the SITARC ontological sensor dataset, the best algorithm is the RF approach with 90.2% accuracy performance. The closest results to the RF approach, DL algorithm with 89.0% accuracy, and DT algorithm with 87.7% accuracy have achieved. Each of these algorithms is considered that suitable for implementation on the SITARC RDF dataset as the prediction model.

Apart from these algorithms, which performed well in terms of accuracy criteria, SVM, GLM, GBT algorithms provided 82.9%, 81.3%, 80.2% accuracy performance, respectively, and proved that they can be used for the SITARC RDF dataset. Among the remaining algorithms, FLM provided an above-average performance with an accuracy of 74.5%. However, this algorithm is thought insufficient for the prediction model, especially when comparing the performance of other algorithms implemented on the SITARC dataset. With the last two approaches, NB and LR algorithms, they had a very bad performance with 36.5% and 16.0% accuracy values, respectively. These two algorithms are thought that considering these results, they cannot be used for the SITARC RDF dataset. The accuracy performances of classical ML algorithms implemented on the SITARC RDF dataset are given in Figure 6.1.

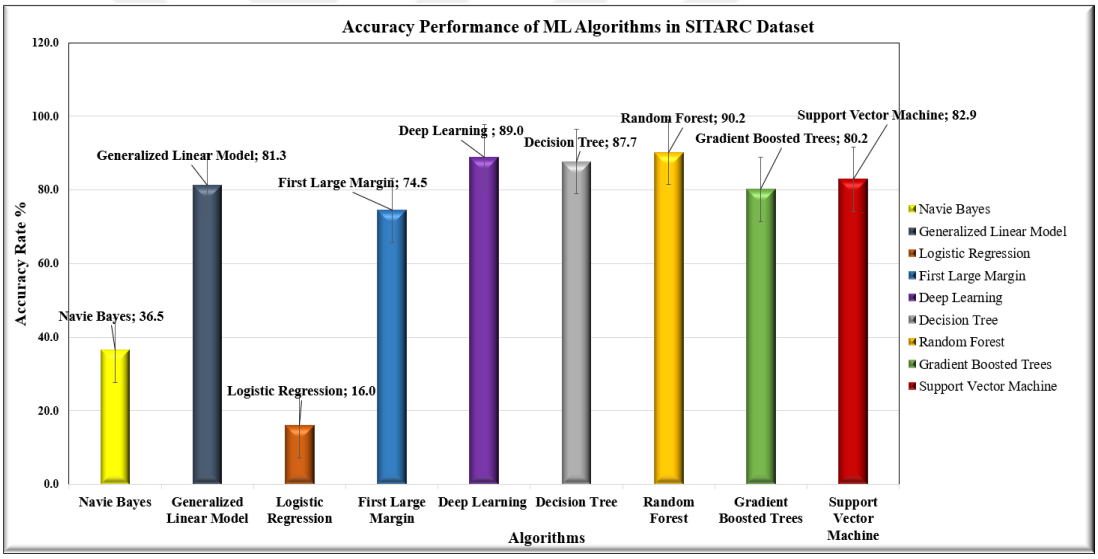


Figure 6.1 Comparison of accuracy of algorithms for SITARC dataset

To be able to say that an algorithm is the best in any dataset, it may not always be sufficient to evaluate it only in terms of accuracy. Especially in proactive systems that can respond in real-time, Training Time and Scoring Time criteria are as important as the accuracy performances of the algorithms, in order to implement the previously planned action plans, as soon as possible and to minimize the loss of possible in unexpected situations.

In order to provide a prediction algorithm that can respond in a reasonable time when applied to any dataset, Training Time and Scoring Time criteria can be improved, sometimes by sacrificing the accuracy of the algorithm. This improvement can generally be done by changing the parameter to be taken by the algorithm or by reducing the number of samples. Some of the algorithms implemented on the SITARC dataset can work with parameters and some without parameters. When the algorithms were first implemented in the SITARC dataset, it was noticed that the Training Time and Scoring Time degree of the tree-based algorithms were at a level that could not compete with other algorithms.

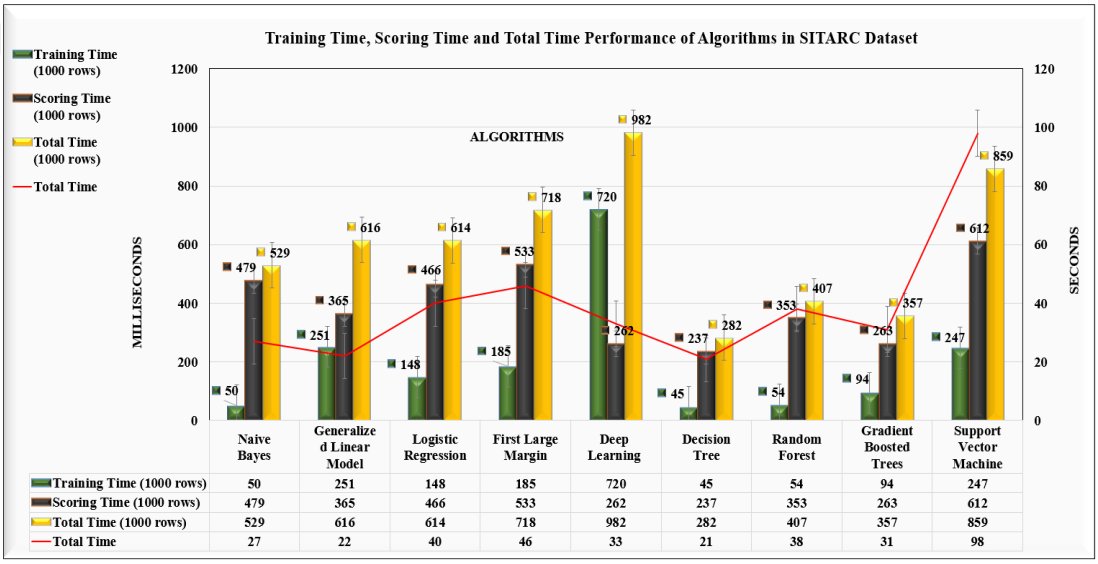


Figure 6.2 Time performance of algorithms implemented in the SITARC dataset

Therefore, some parameters of DT, RF, GBT algorithms, which are in the Tree algorithms family, such as Maximal Depth (MD), Number of Trees (NT), have been changed. Thus, by compromising the accuracy of these algorithms, it was ensured that they could compete with other algorithms in terms both of time as well as accuracy performance. The default parameters of these algorithms that given in RMS Auto Mode, the parameters that are given manually for this dataset, and the working principles of the algorithms are described respectively in terms of the DT algorithm in Section 5.7.2.6, RF algorithm in Section 5.7.2.7, and GBT algorithm in Section 5.7.2.8. In Figure 6.2, the performances of the algorithms implemented on the SITARC RDF dataset in terms of Training Time, Scoring Time, and Total Time are given.

Among the algorithms implemented on this dataset, the best approach in terms of Training Time was the DT algorithm with 45 ms., while the closest performances to this algorithm were shown by NB with 50 ms and RF algorithms with 54 ms. Apart from these algorithms, GBT with 94 ms., LR with 148 ms., and FLM algorithms with 185 ms. showed an above-average Training Time, and they showed that they have the potential to be used even if not the best for SITARC dataset enriched with semantic web technologies. Finally, SVM with 247 ms., GLM with 251 ms., and DL with 720 ms. performed a training time far below the average. According to these results, these algorithms are thought that they were not suitable for the SITARC dataset in terms of Training Time.

However, evaluating only Training Time performance may not be enough, to say that an algorithm has a good performance in terms of time. Algorithms implemented in the dataset should also be evaluated in terms of Scoring Time, which is another time performance indicator. The time evaluation of the algorithms for sensor data should be made by looking at the sum of these two criteria performances. Because, some algorithms implemented in the SITARC dataset have better Training Time, while some algorithms have better Scoring Time.

When the results of the algorithms implemented in the SITARC dataset are considering in terms of Scoring Time, the best algorithm is the DT algorithm with 237 ms., just like in Training Time performance. Following DT's Scoring Time performance, DL with 262 ms. and GBT algorithms with 263 ms., conducted a performance close to the DT algorithm. As seen in Figure 6.2, the DL algorithm, which is the last algorithm in terms of Training Time performance, can show a performance close to the best performance in terms of Scoring Time. Therefore, as mentioned before, it is best to look at the Total Time (Training Time + Scoring Time) performance for the time performance of an algorithm, especially when working with sensor data in an expert system working in real-time.

Considering the Scoring Time performance of other algorithms implemented in the SITARC dataset, the RF algorithm scored the SITARC test dataset with 353 ms., and the GLM algorithm scored with 365 ms. in a reasonable time. Figure 6.2 shows that these algorithms can be used for the SITARC RDF dataset in terms of Scoring Time. Among the remaining approaches, the LR algorithm obtained 466 ms., and the NB algorithm of 479 ms. an average value with Scoring Time performances. However, considering Scoring Time among the algorithms applied to the SITARC dataset, the worst performances were the FLM algorithm with 533 ms. and SVM with 612 ms. These algorithms have been seen obviously that cannot be used for the SITARC RDF dataset when compared to the Scoring Time performances of other approaches.

However, due to the reasons explained before, to say that an algorithm is the best for a dataset in terms of time, it is necessary to refer to Total Time performances, especially in real-time systems (real-time systems where modeling needs to be calculated instantaneously). According to the data in Figure 62, when the approaches implemented to the SITARC dataset, are compared, in terms of Total Time, the DT algorithm has shown the best performance with 21 sec. The algorithms showing the closest performance to the DT algorithm in terms of Total Time are the GLM algorithm with 22 sec and the NB algorithm with 27 sec. In terms of Total Time criteria, these algorithms are considered that may be used in proactive systems as prediction models, for the SITARC RDF dataset.

When the results are evaluated in terms of Total Time performance, they showed close to the average performance value the GBT algorithm with 31 sec., the DL algorithm with 33 sec., and the RF algorithm with 38 sec. after these algorithms. These approaches have proven that they can be used for the SITARC dataset if they give very good results in terms of accuracy. In terms of Total Time performance, among the remaining algorithms, the LR algorithm with 40 sec., and the FLM algorithm with 46 sec. As seen in Figure 6.2, when the algorithms applied to the SITARC RDF dataset are evaluated in terms of Total Time results, the approach with the worst performance was definitely the SVM algorithm with a degree of 98 sec.

In Figure 6.3, the graph of the correlations between the attributes in the SITARC dataset and the result variable is given. These correlation values are actually a numerical representation of how effective that attribute has on the result.

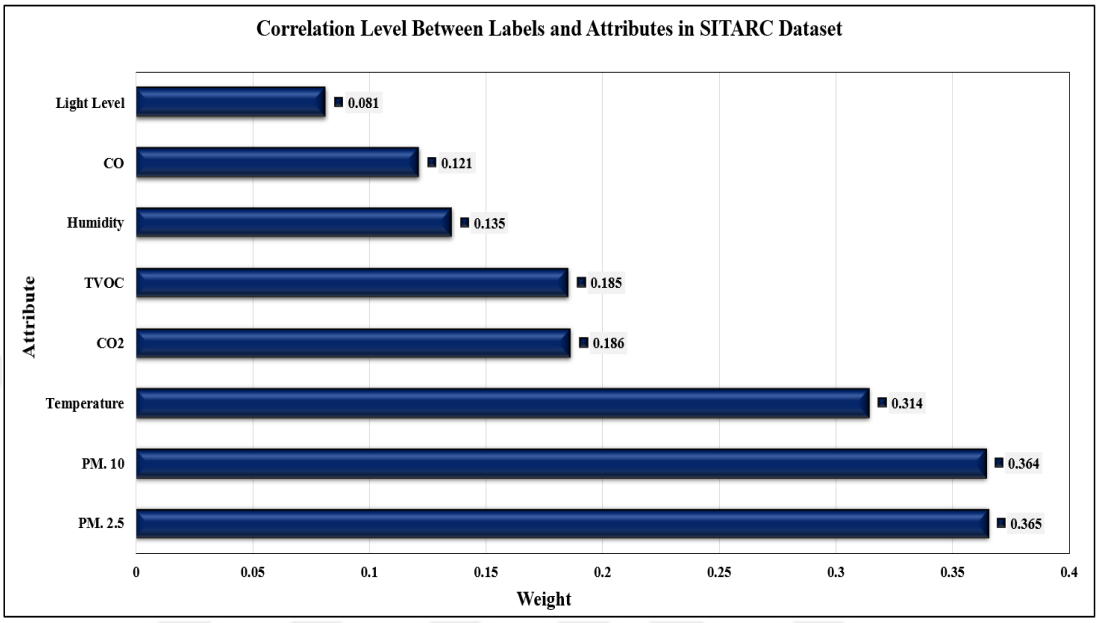


Figure 6.3 Correlation of the attributes in the SITARC dataset

According to the data in Figure 6.3, it is seen that the attributes that affect the result label the most are $PM_{2.5}$ with 0.365 weight value and PM_{10} with 0.364 weight value. Another attribute that greatly influenced the result was the Temperature parameter with a value of 0.314 weight. When previous studies were taken as reference, it was predicted that $PM_{2.5}$ and PM_{10} attributes would stand out among the parameters of indoor air quality. But it was impossible to predict that the temperature parameter could affect the result so much. Apart from these attributes, the correlation between the result label and the CO_2 parameter with 0.186 weight value, and the TVOC parameter with 0.185 weight value, remained at an average value. As seen in Figure 6.3, although the Humidity parameter with 0.135 weight value and the CO parameter with 0.121 weight value does not affect the result label much, that the algorithms are important parameters in terms of creating a model on the SITARC dataset. Finally, the light level parameter with 0.081 weight value does not have a significant effect on the result is seen clearly.

Algorithms implemented in any dataset comparison of the criteria such as in terms of Accuracy, Training Time, Scoring Time, Flexibility, etc. is sometimes maybe not enough for selecting the prediction model. Especially if there is more than one parameter affecting the result label in the dataset and the result label is classified in more than two categories, it may be useful to calculate gain value while comparing algorithms to produce more effective solutions in decision-making processes. In the proposed thesis study, while the SITARC dataset consists of 8 attributes, the result label is divided into 5 different categories to push the limitations of the algorithms.

Therefore, within the scope of the proposed thesis, the algorithms implemented in the datasets were evaluated in terms of accuracy and time, as well as in terms of gain and loss. Because, in a multi-class dataset such as the SITARC dataset, the predicted value is in a class close to the true value is more acceptable than it is in a class far from the true value. A cost matrix was created to evaluate algorithms in terms of gain and loss. While creating this Cost matrix, the distance of the estimated value from the actual values was taken as a reference. An example is given in Figure 6.4 to make the matrix used when calculating the gain and loss easier to understand.

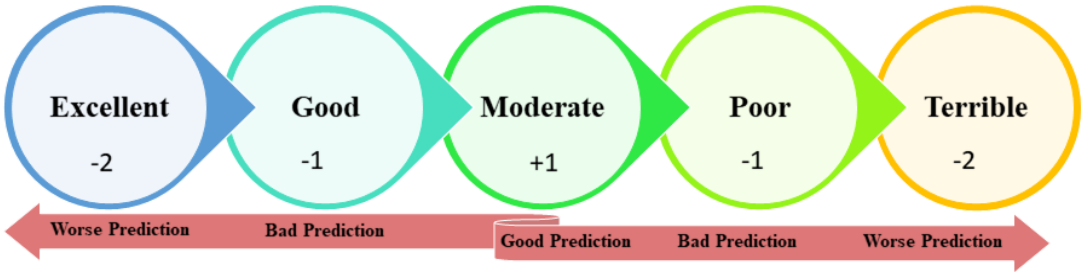


Figure 6.4 The sample explaining how to create a cost matrix

Suppose it is an example whose real value is given as "Moderate" as seen in Figure 6.4. If the estimated value is "Moderate", the gain of the algorithm in that example is calculated as +1. Any other estimation means a loss for the algorithm. If the predicted value is in a neighborhood of the actual value, like "Poor" and "Good" classes, the algorithm's loss for that instance is-1. Finally, if the predicted value is in the two neighborhoods of the actual value, like "Excellent" and "Terrible" classes, the algorithm's loss for that example is taken as-2. Cost Matrix was created using logic in

this example. Thus, the gain value in Cost Matrix takes a variable value between -4 and +1.

Table 6.1 Cost matrix referenced when comparing the gain of algorithms

Cost Matrix	True Terrible	True Poor	True Moderate	True Good	Ture Excellent
Predicted Terrible	1	-1	-2	-3	-4
Predicted Poor	-1	1	-1	-2	-3
Predicted Moderate	-2	-1	1	-1	-2
Predicted Good	-3	-2	-1	1	-1
Predicted Excellent	-4	-3	-2	-1	1

The gains and costs of incorrect and correct estimates are given in Table 6.1. Losses are represented by negative numbers while gains are represented by positive numbers. A detailed example is given in Table 6.2 to better understand the use of Cost Matrix when comparing the performance of algorithms.

Table 6.2 An Example of cost matrix use

Actual Label Value	Prediction Label Value	Cost/Loss Value	Accuracy
Excellent	Terrible	-4	FALSE
Excellent	Poor	-3	FALSE
Excellent	Moderate	-2	FALSE
Excellent	Good	-1	FALSE
Excellent	Excellent	1	TRUE

According to Table 6.2, for example, if the label value of an instance whose actual label value is “Excellent” is estimated as “Excellent” with any classifier, the prediction is correct and takes 1 as the gain point. On the other hand, if the classifier labeled the same “Excellent” instance as a “Good”, “Moderate”, “Poor”, or “Terrible” the classifier takes -1, -2, -3, -4 loss point respectively and this prediction becomes wrong. These loss points give the value of the wrong prediction. In some cases, especially

multi-label and multi-attribute datasets, it may be more beneficial to choose the best performing algorithm by looking at gain rather than accuracy. The gain performance of the algorithms implemented in the SITARC RDF dataset is given in Figure 6.5.

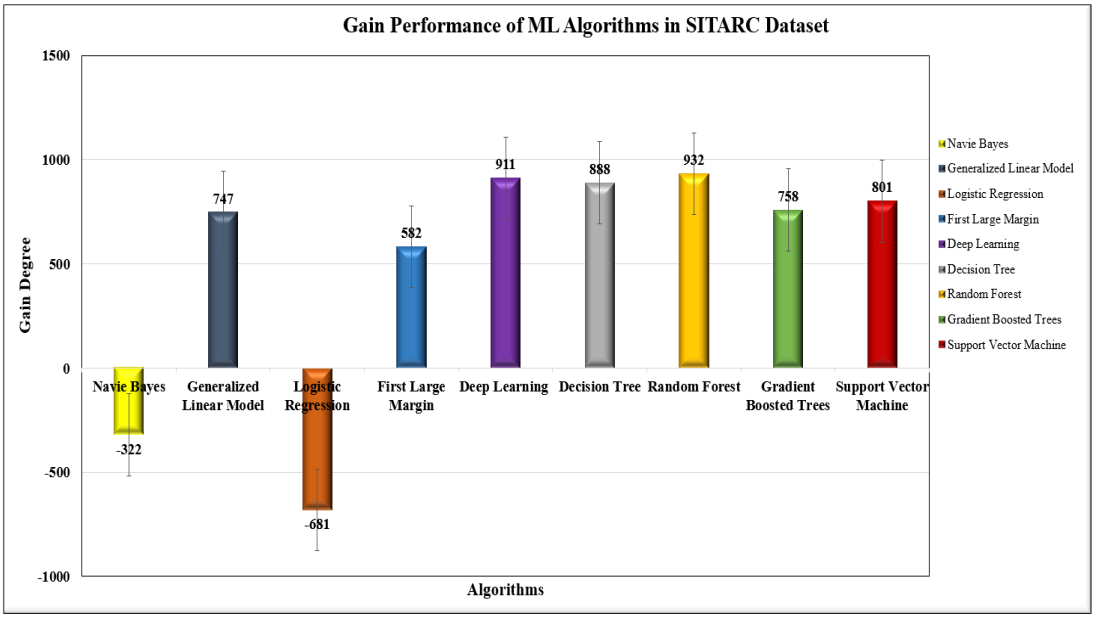


Figure 6.5 Comparison of gain performance of algorithms for SITARC database

When the performances of the algorithms are compared in terms of gain, it is seen that the sum of the costs of NB and LR algorithms is negative, while the remaining algorithms are positive. Parallel to the accuracy performance, it is seen in Figure 6.5 that the RF approach provides the best performance with 932 scores. Following the RF algorithm, it is seen that DL with 911 score and DT approaches with 888 score come. These 3 algorithms have proven with these scoring their usability for the SITARC dataset enriched with semantic web technologies in terms of gain.

Apart from these algorithms, the SVM approach with 801 scores, the GBT approach with 758 scores, and the GLM approach with 747 scores achieved an above-average gain score in the SITARC dataset. The FLM algorithm with 582 score is considered that cannot be used as a prediction approach in a proactive system design in the SITARC dataset by obtaining a score close to the average. The remaining NB and LR obtained results that were incomparably worse than other algorithms, with -322 and -681 scores, respectively.

When the performances of 9 classical ML algorithms implemented in the SITARC RDF dataset are analyzed in terms of Accuracy, Training Time, Scoring Time, Total Time, and Gain; The RF, DT, and DL algorithms which are stood out among others. These algorithms are thought that they can be used in a proactive system for a prediction approach by looking at the above graphs and comments. Especially, although it has provided an accuracy rate of 89%, the DL algorithm should not be forgotten to maybe insufficient for a system operated in real-time, in terms of Total Time performance. Among the remaining two approaches, the DT algorithm provided about two times better performance than the RF algorithm in terms of Total Time. Therefore, the project team's recommendation is to use the DT algorithm with a little compromise on accuracy in vital processes that require instant analysis in the SITARC RDF dataset, and where possible action plans need to be processed quickly.

6.3 Performance of Classical ML Algorithms on the MICU Dataset

First, the results of classical ML algorithms implemented on the SITARC dataset are presented above. After that, in order to evaluate the performance of the same classical ML algorithms in semantically enriched datasets, the MICU RDF dataset was also implemented and compared. As a result of all the comparisons, when the MICU RDF dataset enriched with semantic web technologies is considered alone, the accuracy performances of the approaches that can be used for prediction in this dataset may be possible to summarize as follows; DL approach was the best algorithm in MICU ontological sensor dataset with 68.1% accuracy performance. Following this algorithm, the FLM algorithm and the GLM algorithm performed above average with an accuracy performance of 62.7% and 58.2% respectively. Each of these algorithms is partially suitable in implementation on the MICU RDF dataset as the prediction model.

As can be seen from the results, the accuracy performance of classical ML algorithms in the MICU dataset is lower than the performance in the SITARC dataset. The main reason for this difference in accuracy performance is definitely the

characteristics of the two datasets. While the classes in the SITARC dataset are distinctly different from each other, the classes in the MICU dataset are sometimes intertwined and difficult to distinguish. The biggest reasons for this can be listed as the disruption of data collection in the MICU environment due to the pandemic, the presence of too many people in the environment and errors due to human curiosity, and the presence of more imputed values in the MICU dataset than the SITARC dataset.

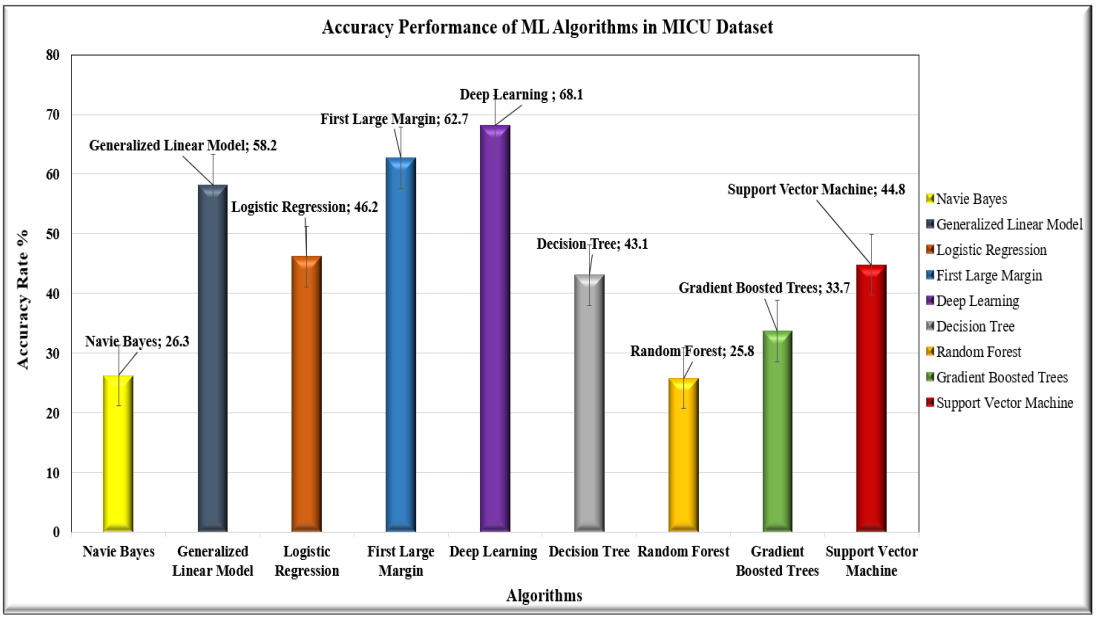


Figure 6.6 Comparison of accuracy of algorithms for MICU database

Apart from these algorithms whose value exceeded the average, the LR algorithm with 46.2%, the SVM algorithm with 44.8%, and the DT algorithm with 43.1% showed a performance close to the average in terms of accuracy performance. However, the GBT algorithm with 33.7%, NB with 26.3%, and RF algorithm with 25.8% conducted a performance below average in terms of accuracy performance. These algorithms are thought that should not be used in the MICU RDF database as a prediction approach. The accuracy performances of classical ML algorithms implemented on the MICU RDF dataset enriched with semantic web technologies are given in Figure 6.6.

To be able to say that an algorithm is the best in any dataset, it may not always be sufficient to evaluate it only in terms of accuracy. Especially in proactive systems that

can respond in real-time, Training Time and Scoring Time criteria are as important as the accuracy performances of the algorithms, in order to implement the previously planned action plans, as soon as possible and to minimize the loss of possible in unexpected situations.

In order to provide a prediction algorithm that can respond in a reasonable time when applied to any dataset, Training Time and Scoring Time criteria can be improved, sometimes by sacrificing the accuracy of the algorithm. This improvement can generally be done by changing the parameter to be taken by the algorithm or by reducing the number of samples. Some of the algorithms implemented on the SITARC and MICU datasets can work with parameters and some without parameters. When the algorithms were first implemented in the MICU dataset, it was noticed that the Training Time and Scoring Time degree of the tree-based algorithms were at a level that could not compete with other algorithms, just like in the SITARC dataset.

Therefore, some parameters of DT, RF, GBT algorithms, which are in the Tree algorithms family, such as MD, NT, have been changed, just like in the SITARC dataset. Thus, by compromising the accuracy of these algorithms, they could compete with other algorithms both in terms of time as well as accuracy performance was ensured in MICU dataset. The default parameters of these algorithms that given in RMS Auto Mode, the parameters that are given manually for this dataset, and the working principles of the algorithms are described respectively in terms of the DT algorithm in Section 5.7.2.6, RF algorithm in Section 5.7.2.7, and GBT algorithm in Section 5.7.2.8. In Figure 6.7, the performances of the algorithms implemented on the MICU RDF dataset in terms of Training Time, Scoring Time, and Total Time are given.

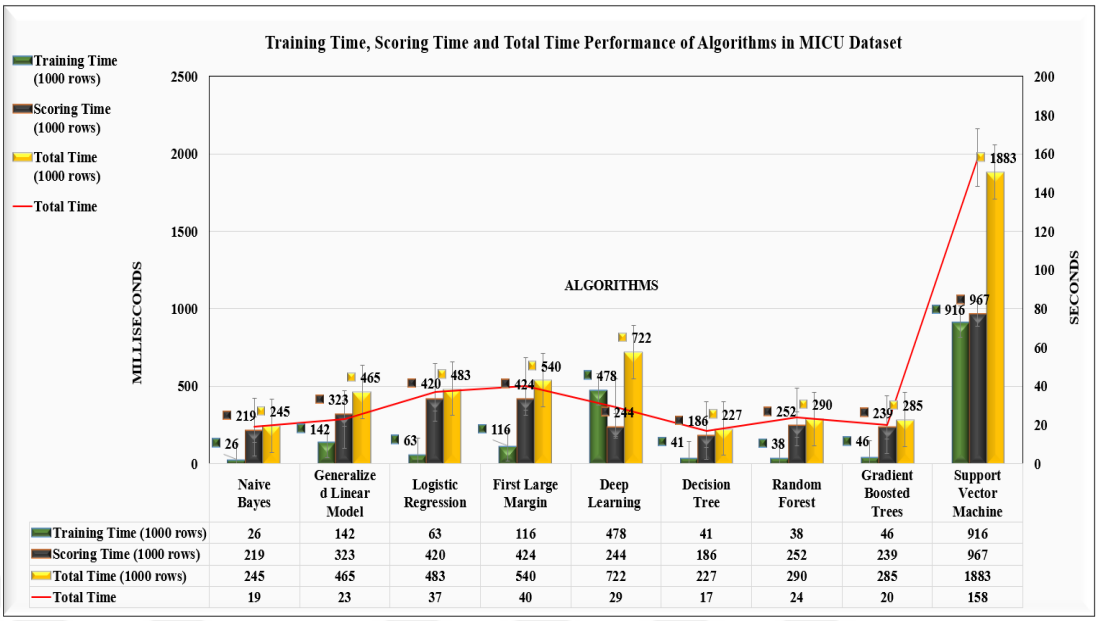


Figure 6.7 Time performance of algorithms implemented in the MICU dataset

Among the algorithms implemented on the MICU RDF dataset, the best approach in terms of Training Time was the NB algorithm with 26 ms., while the closest performances to this algorithm conducted the RF algorithm with 38 ms., the DT algorithm with 41 ms., the GBT algorithm with 46 ms., and the LR algorithm with 63 ms. These approaches have shown a timing close to ideal in terms of Training Time, showing that they can be used for the MICU dataset. Apart from these algorithms, FLM algorithms with 116 ms. and GLM algorithms with 142 ms. show a training time close to the average, while they are thought to have the potential to be used even if not the best for MICU dataset enriched with semantic web technologies. Finally, considering Training Time among the algorithms applied to the MICU dataset, the worst performances were the DL algorithm with 478 ms., the SVM algorithm with 967 ms. These algorithms are thought that they are not suitable for the MICU RDF dataset in terms of the Training Time.

However, as mentioned earlier, evaluating only Training Time performance may not be enough, to say that an algorithm has a good performance in terms of time. Algorithms implemented in the dataset should also be evaluated in terms of Scoring Time, which is another time performance indicator. The time evaluation of the algorithms for sensor data should be made by looking at the sum of these two criteria

performances. In the MICU dataset, just like in the SITARC dataset, some algorithms implemented have better Training Time, while some algorithms have better Scoring Time.

When the results of the algorithms implemented in the MICU dataset are analyzed in terms of Scoring Time, the best algorithm is the DT algorithm with 186 ms., just like in Training Time performance in the SITARC dataset. Following DT's Scoring Time performance, the NB algorithm with 219 ms., the GBT algorithm with 239 ms., the DL algorithm with 244 ms., and the RF algorithm with 252 ms. conducted a performance close to the DT algorithm. As seen in Figure 6.7, the DL algorithm, which performs well below the average in terms of Training Time performance, can perform well above the average in terms of Scoring Time.

The scoring Time performance of the GLM algorithm, which is another approach implemented to the MICU dataset, is close to the average value with 323 ms. However, while there are much better performances in terms of Scoring time, it is a little unlikely that this algorithm will be chosen as the prediction model for a proactive system. Considering the performance of other algorithms implemented in the MICU dataset in terms of Scoring Time, the LR algorithm with 420 ms. and the FLM algorithm with 424 ms. scored the MICU test dataset in a below-average time. In terms of Scoring Time, these algorithms have been seen obviously that cannot be used for the MICU RDF dataset. Finally, considering Scoring Time among the algorithms applied to the MICU dataset, the worst performance was the SVM algorithm with 967 ms. This algorithm has been thought that cannot be used for the MICU dataset enriched with semantic web technologies compared to the Scoring Time performances of other approaches.

However, due to the reasons explained before, to say that an algorithm is the best for a dataset in terms of time, it is necessary to consider Total Time performances, especially in real-time systems. When the approaches implemented to the MICU dataset are compared in terms of Total Time, according to the data in Figure 6.7, the DT algorithm showed the best performance with 17 sec., just like in the SITARC

dataset. The algorithms showing the closest performance to the DT algorithm in terms of Total Time are NB with 19 sec., GBT with 20 sec., GLM with 23 sec., and RF with 24 sec. Considering Total Time, these algorithms are thought that can be used as a prediction approach in a proactive system design in the MICU RDF dataset.

When the results were evaluated in terms of Total Time performance, DL with 29 sec., LR with 37 sec., and FLM with 40 sec. after these algorithms showed a performance above the average value. In these approaches, they have proven that if they give very good results in terms of accuracy, they can be used for the MICU dataset. As seen in Figure 6.7, when the algorithms applied to the MICU RDF dataset are evaluated in terms of Total Time results, the approach with the worst performance was definitely SVM algorithm with a degree of 158 sec., as in the SITARC dataset.

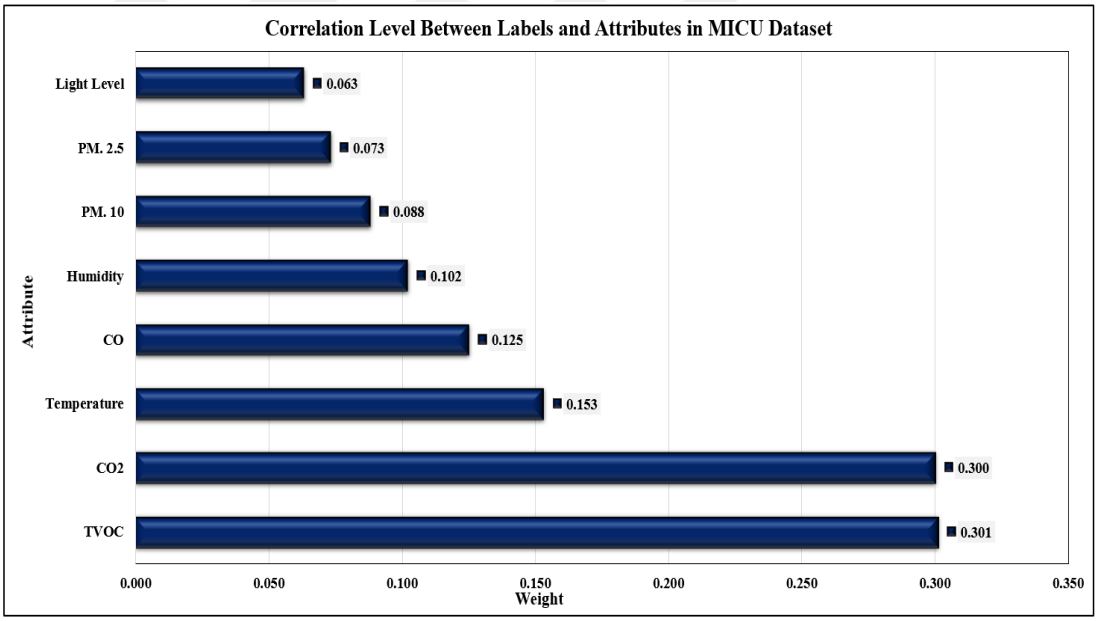


Figure 6.8 Correlation of the attributes in the MICU dataset

In Figure 6.8, the graph of the correlations between the attributes in the MICU dataset and the result variable is given. These correlation values are actually a numerical representation of how effective that attribute has on the result. According to the data in Figure 6.8, unlike the SITARC dataset, the attribute that affects the result label the most is TVOC with a value of 0.301 weight and CO_2 with a value of 0.300 weight, respectively. The fact that $PM_{2.5}$ and PM_{10} values have so little effect on the

result label can be attributed to the fact that an external hepa filter is used in the hospital intensive care unit. However, it is up to the managers to decide whether the $PM_{2.5}$ and PM_{10} levels, which are vital for the health of the patients in a place that has critical importance such as an intensive care unit, are sufficient.

One of the other attributes that greatly influenced the result was Temperature with a value of 0.153 weight and CO parameters with a value of 0.125 weight. Apart from these attributes, with 0.102 weight value, the correlation between the result label and the Humidity parameter is slightly below the average. With 0.88 weight value PM_{10} and with 0.73 weight value $PM_{2.5}$ attributes were not affecting the result label much, but it is certain that classical ML algorithms are important parameters in terms of creating models on the MICU dataset. Finally, it is clearly seen in Figure 6.8 that with its 0.63 weight value, the light level parameter does not have a significant effect on the result.

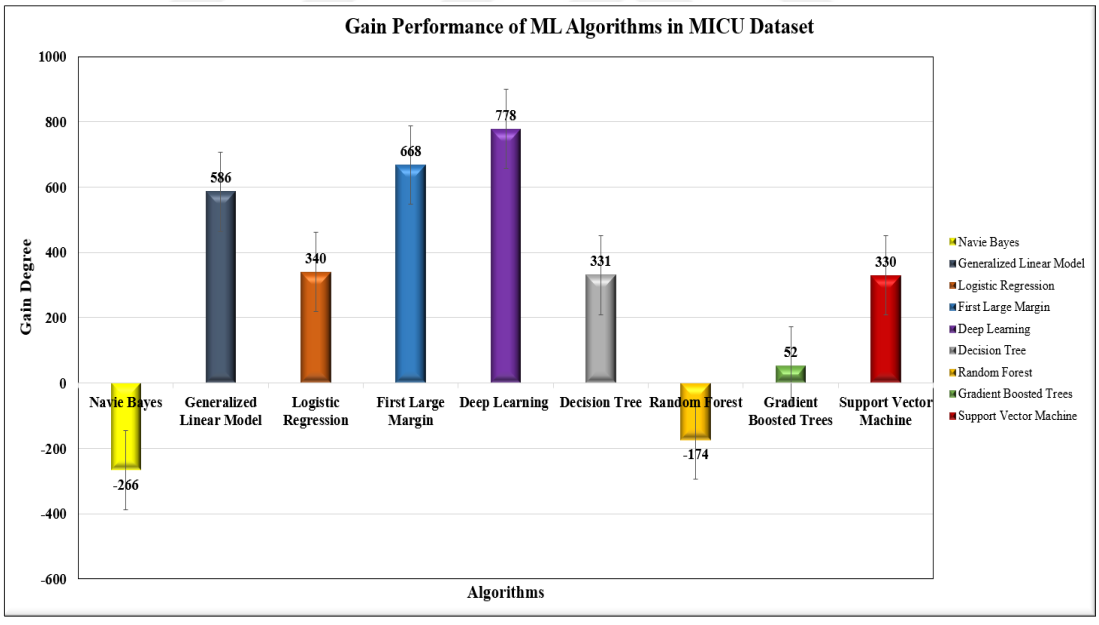


Figure 6.9 Comparison of gain performance of algorithms for MICU database

Algorithms implemented in any dataset comparison of the criteria such as in terms of Accuracy, Training Time, Scoring Time, Flexibility, etc. is sometimes maybe not enough for selecting the prediction model. Especially if there is more than one parameter affecting the result label in the dataset and the result label is classified in

more than two categories, it may be useful to calculate gain value while comparing algorithms to produce more effective solutions in decision-making processes. In the proposed thesis, the MICU dataset consists of 8 attributes like the SITARC dataset, while the result label is divided into 5 different categories to push the limits of the algorithms.

Therefore, within the scope of the proposed thesis, the algorithms implemented on the MICU datasets were evaluated in terms of accuracy and time, as well as in terms of gain and loss. Because, in a multi-class dataset such as the MICU dataset, the predicted value is in a class close to the true value is more acceptable than it is in a class far from the true value. A cost matrix was created to evaluate algorithms in terms of gain and loss. While creating this Cost matrix, the distance of the estimated value from the actual values was taken as a reference. The gain performance of the algorithms implemented in the MICU RDF dataset is given in Figure 6.9.

Cost Matrix used for gain calculation in MICU dataset is the same as Cost matrix used in SITARC dataset. For this reason, how to create the Cost matrix will not be explained again in this section. Required information about Cost Matrix; An example is given in Figure 6.4 to make it easier to understand the matrix used when calculating the gain and loss. The benefits and costs of incorrect and correct estimates are given in Table 6.1. An example is given in Table 6.2 to better understand the use of Cost Matrix when comparing the performance of algorithms.

When the performances of the algorithms are compared in terms of gain, it is seen that the sum of the costs of NB and RF algorithms is negative, while the remaining algorithms are positive. Figure 6.9 reveals that the DL approach provides the best performance with 778 scores in parallel with the accuracy performance when the performances of algorithms are analyzed in terms of gain. The algorithms following the performance of the DL algorithm, are the FLM approaches with 668 scores and the GLM approaches with 586 scores. It is thought that these 3 algorithms can be used in terms of gain because they are higher than the average value for the MICU dataset enriched with semantic web technologies.

Apart from these algorithms, the LR algorithm with 340 score, DT algorithm with 331 score, and the SVM algorithm with 330 score obtained an average gain score in the MICU dataset. However, it is obvious that these gain scores are insufficient for the prediction model required for a proactive system design. Following these approaches, it is certain that the GBT algorithm with a score of 52 points cannot be used as a prediction approach in a proactive system design in the MICU dataset. RF and NB algorithms achieved poor results that could not be compared with other algorithms, while they obtained -174 and -266 scores, respectively. Experimental results show that these algorithms not suitable for the MICU dataset as a prediction model.

When the performances of 9 classical ML algorithms implemented in the MICU RDF dataset are analyzed in terms of Accuracy, Training Time, Scoring Time, Total Time and Gain; The DL, FLM, and GLM algorithms which are stood out among others. These algorithms are thought that they can be partially used in a proactive system for a prediction approach by looking at the above graphs and comments. There are multiple reasons why the algorithm performances are lower in the MICU dataset compared to the SITARC dataset. These reasons will be discussed in Chapter 7 conclusions. However, it should not be forgotten that although the DL algorithm provides the best accuracy score, it may be insufficient for a system operated in real-time especially the model generation time.

In terms of Total Time, the GLM algorithm provided better timing performance than the FLM algorithm among the remaining two approaches. However, the accuracy performances of these algorithms are slightly lower than DL. Although the Training Time of the DL algorithm is high, experimental results have shown that the performance of DL better than the average of the FLM and GLM algorithms when the Total Time is examined by closing this gap during the application of the model to the dataset. Therefore, the project team's recommendation is to use the DL algorithm, which is more reasonable in accuracy performance with little compromise in time, in vital processes that require instant analysis in the MICU RDF dataset and where possible action plans need to be executed quickly.

6.4 Performance of Classical ML Algorithms on the COMBINED Dataset

The results of the experiments performed on the SITARC and MICU dataset and the comparison of the results of 9 classical ML algorithms applied are presented supported by graphical data from different angles in this section. In addition, the SITARC dataset created from the IAQ parameters collected in the laboratory environment and the MICU datasets created from the IAQ parameters collected from the hospital intensive care unit will be combined and the experiments performed in this section have been implemented into this COMBINED RDF ontological sensor dataset.

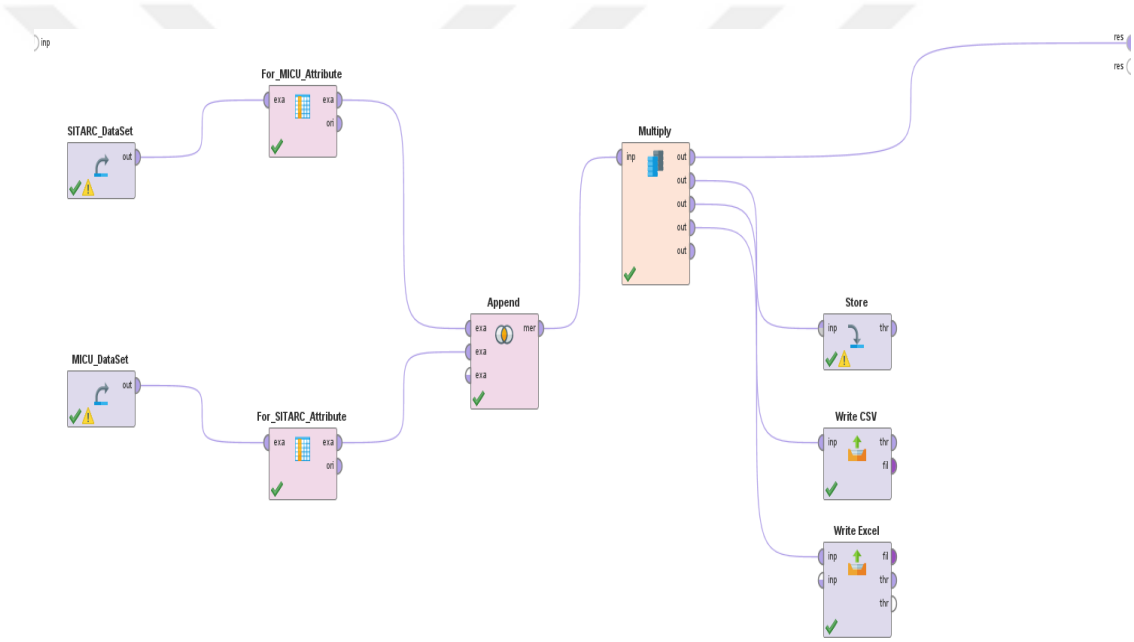


Figure 6.10 Schematic representation of the creation of the COMBINED dataset

When determining the appropriate prediction algorithm on datasets enriched with semantic web technologies, implementing algorithms that determined previously, in different datasets is very important in or order to compare and check results. It is thought that testing the approaches in many datasets may be useful in determining the most appropriate prediction model. For this reason, a COMBINED RDF dataset was created by combining SITARC and MICU datasets to create a different perspective. The merging of SITARC and MICU RDF datasets is provided with the Append

operator in RMS. Creation of COMBINED dataset in RMS environment is given schematically in Figure 6.10.

The append operator was used to create the COMBINED RDF dataset shown in Figure 6.10. However, while combining SITARC and MICU datasets, some problems arose due to different parameters. These problems were solved by clearing the columns that make up the two datasets with the Select Attribute operator without coming to the Append operator. The newly created COMBINED RDF dataset was stored in the RMS environment with the STORE operator for model selection. In addition, the newly created COMBINED RDF dataset was stored in CSV format with the "Write CSV" operator and in XSL format with the "Write Excel" operator for use in other environments.

The results of classical ML algorithms, which were first implemented on the SITARC and MICU dataset, are presented above. After that, the same classical ML algorithms were implemented to the COMBINED RDF dataset created by combining the semantically enriched SITARC and MICU datasets in order to gain a different perspective and enrich the results obtained. Results from this experiment compared with each other and analyzed. As a result of all the comparisons, when the COMBINED RDF dataset enriched with semantic web technologies is considered alone, it may be possible to summarize the accuracy performance of the approaches that can be used for estimation for this dataset as follows.

The best algorithm in the COMBINED ontological sensor dataset was the DL approach with 79.8% accuracy performance as in the MICU dataset. Following this algorithm, the RF algorithm showed an accuracy performance of 79.0%, the DT algorithm with 75.4%, and the SVM algorithm with an accuracy of 73.1, showing a performance close to the DL approach, which shows the best performance. Each of these algorithms is considered that suitable in terms of accuracy performance for the prediction model in the COMBINED RDF dataset.

Apart from these algorithms at the top ranks, the FLM algorithm with an accuracy performance of 69.7%, the GLM algorithm with an accuracy performance of 68.0%, and the LR algorithm with an accuracy performance of 61.4% also performed above average. Although these algorithms have above average accuracy, they are insufficient to be used as a prediction approach in a proactive system design. Among the algorithms implemented in the COMBINED dataset, GBT, and NB algorithms conducted an accuracy performance of 54.0% and 49.5% respectively. Data in Figure 6.11 shows that the accuracy performance of these algorithms is well below the average performance of all approaches. For this reason, it is thought that these approaches should not be used in the COMBINED RDF database, while there are algorithms that have better accuracy performances. The accuracy performance of classical ML algorithms implemented on the COMBINED RDF dataset enriched with semantic web technologies is given in Figure 6.11.

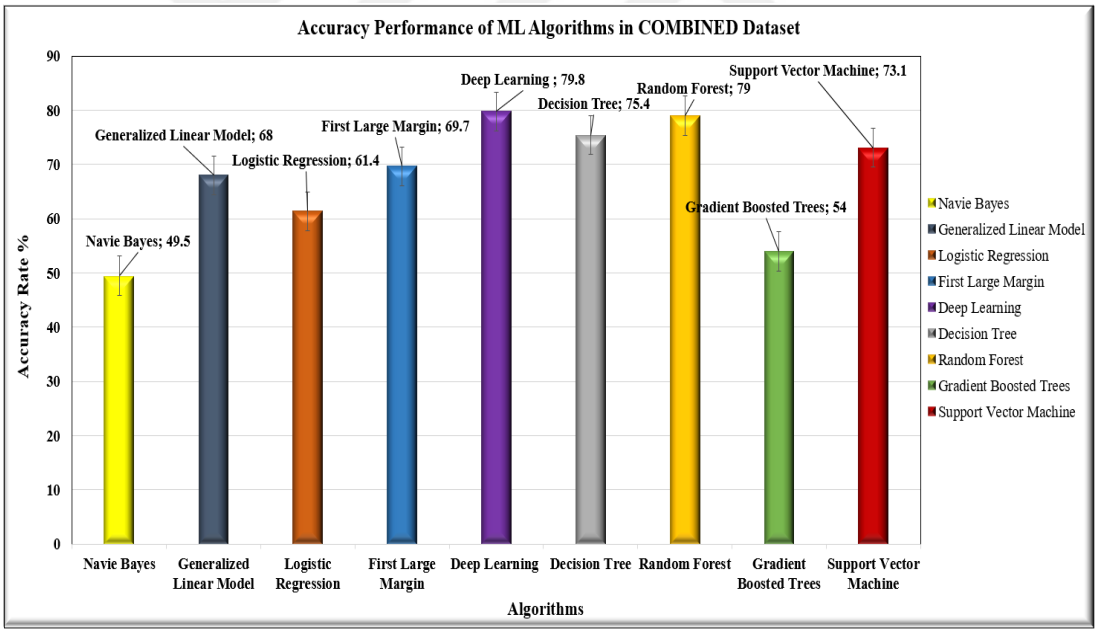


Figure 6.11 Comparison of accuracy of algorithms for COMBINED database

To be able to say that an algorithm is the best in any dataset, it may not always be sufficient to evaluate it only in terms of accuracy. Especially in proactive systems that can respond in real-time, Training Time and Scoring Time criteria are as important as the accuracy performances of the algorithms, in order to implement the previously planned action plans, as soon as possible and to minimize the loss of possible in

unexpected situations. For this reason, the results of ML approaches used in the COMBINED dataset, just like in the SITARC and MICU datasets, were compared in terms of model creation time, test time, and Total Time, and the results were shared. In Figure 6.12, the performances of the algorithms implemented on the COMBINED RDF dataset in terms of Training Time, Scoring Time, and Total Time are given.

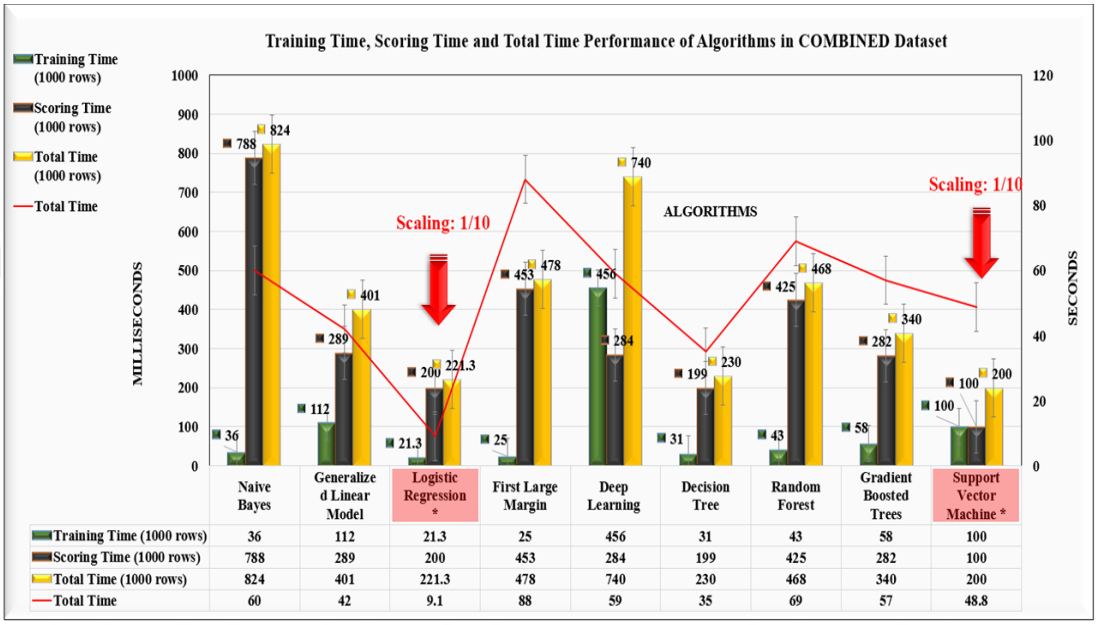


Figure 6.12 Time performance of algorithms implemented in the MICU dataset

In order to provide a prediction algorithm that can respond in a reasonable time when applied to any dataset, Training Time and Scoring Time criteria can be improved, sometimes by sacrificing the accuracy of the algorithm. This improvement can generally be done by changing the parameter to be taken by the algorithm or by reducing the number of samples. Just like in SITARC and MICU datasets, some of the algorithms implemented on the COMBINED datasets can work with parameters and some without parameters. When the algorithms were first implemented in the COMBINED dataset, it was noticed that the Training Time and Scoring Time degree of the tree-based algorithms were at a level that could not compete with other algorithms.

Therefore, some parameters of DT, RF, GBT algorithms, which are in the Tree algorithms family, such as MD, NT, have been changed, just like in the SITARC and

MICU dataset. Thus, by compromising the accuracy of these algorithms, they could compete with other algorithms both in terms of time as well as accuracy performance was ensured in COMBINED dataset.

Among the algorithms implemented on the COMBINED RDF dataset, the best approach in terms of Training Time was the FLM algorithm with 25 ms., while the closest performances to this algorithm were conducted respectively by the DT algorithm with 31 ms., the NB algorithm with 36 ms., the RF algorithm with 43 ms., and the GBT algorithm with 58 ms. These approaches showed a timing close to ideal for the COMBINED dataset in terms of Training Time. This experiment results show that they can be used for this dataset. Apart from these algorithms, GLM algorithms with 112 ms. and LR algorithms with 213 ms. showed an above-average Training Time, and they showed that they have the potential to be used even if not the best for MICU dataset enriched with semantic web technologies.

DL algorithm showed an average performance with a Training Time of 478 ms. Although the DL approach has an average performance, it is thought that this algorithm can be used as a prediction approach in cases where it can be more successful than other algorithms in terms of accuracy. Finally, with a performance above 1,000 ms., the SVM algorithm showed a Training Time far below average, showing that it is not suitable in terms of the Training Time for the COMBINED RDF dataset. (Values taken by SVM and LR algorithms in terms of time criteria are scaled as 1/10 for better understanding of the graph.)

However, as mentioned earlier, evaluating only Training Time performance may not be enough, to say that an algorithm has a good performance in terms of time. Algorithms implemented in the dataset should also be evaluated in terms of Scoring Time, which is another time performance indicator. The time evaluation of the algorithms for sensor data should be made by looking at the sum of these two criteria performances. In the COMBINED dataset, just like in the SITARC and MICU dataset, some algorithms implemented have better Training Time, while some algorithms have better Scoring Time. This difference is due to the fact that the working principles of

the algorithms are completely different from each other. While some algorithms make a lot of effort in creating the model, some algorithms take a lot of effort while scoring the test dataset.

When the results of the algorithms implemented in the COMBINED dataset are checked in terms of Scoring Time, it is seen that the best algorithm is the DT algorithm with 199 ms. Following DT's Scoring Time performance, the performance of GBT algorithms with 282 ms., DL with 284 ms., and GLM with 289 ms. is listed. As seen in Figure 6.12, the DL algorithm, which has an average performance in terms of Training Time performance, can perform above average in terms of Scoring Time just like in SITARC and MICU datasets.

Scoring Time performance of RF and FLM algorithms, other approaches implemented to the COMBINED dataset, is close to the average value with 425 ms. and 453 ms., respectively. However, while there are much better performances in terms of Scoring time, it is a little unlikely that this algorithm will be chosen as the prediction model for a proactive system. The performance of the NB approach, which is another algorithm implemented into the COMBINED RDF dataset, is 788 ms. in terms of Scoring Time.

This value, which is below the average scoring time performance, is insufficient in terms of using the NB algorithm for the COMBINED dataset. Finally, considering Scoring Time among the algorithms applied to the COMBINED dataset, the worst performances were SVM with performance over 1,000 ms. and LR algorithms with a performance above 2,000 ms. Experimental results show these algorithms cannot be used for the semantic enriched COMBINED dataset with semantic web technologies when compared to the Scoring Time performances of other approaches.

According to the data in Figure 6.12, the DT algorithm has shown the best performance with 35 sec., just like the SITARC and MICU datasets when the approaches implemented to the COMBINED dataset are compared in terms of Total Time. The algorithms showing the closest performance to the DT algorithm in terms

of Total Time are the GLM algorithm with 42 sec., the GBT algorithm with 57 sec., the DL algorithm with 59 sec., and the NB algorithm with 60 sec. Considering Total Time in these algorithms, it is thought that COMBINED can be used as a prediction approach for a proactive system design for the RDF dataset.

When the results were evaluated in terms of Total Time performance, they showed a performance above the average value of the RF algorithm with 69 sec., the FLM algorithm with 88 sec., and the LR algorithm with 91 sec. after these algorithms. In these approaches, they have proven that if they give very good results in terms of accuracy, they can be used for the COMBINED dataset. As seen in Figure 6.12, when the algorithms applied to the COMBINED RDF dataset are evaluated in terms of Total Time results, the approach with the worst performance was definitely the SVM algorithm with a 488 sec. rating, as in the SITARC and MICU dataset. Even if the accuracy of this approach is better than other algorithms, it is thought that it should not be used as a prediction model because it would not respond in a reasonable time in a real-time system.

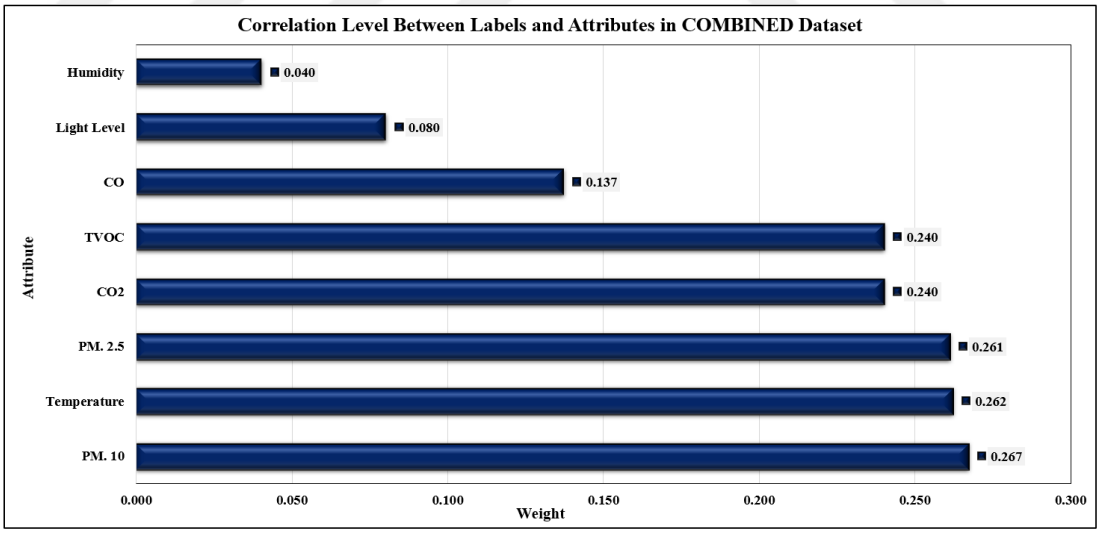


Figure 6.13 Correlation of the attributes in the COMBINED dataset

In Figure 6.13, the graph of the correlations between the attributes in the COMBINED dataset and the result variable is given. These correlation values are actually a numerical representation of how effective that attribute has on the result. According to the data in Figure 6.13, unlike the SITARC and MICU datasets, the

attribute that affects the result label the most is PM_{10} with a value of 0.267 weight, Temperature with a value of 0.262 weight, and $PM_{2.5}$ with a value of 0.261 weight respectively.

One of the other attributes that affect the result at least as much as these attributes are CO_2 with 0.240 weight value and TVOC with 0.240 weight value. These 5 attributes had a more significant effect on the COMBINED RDF dataset than other attributes and played an important role in determining the characteristics of this dataset. Apart from these attributes, the correlation between CO and the result label with a value of 0.137 weight was slightly below average. Light Level with 0.80 weight value and Humidity attributes with 0.40 weight value does not affect the result label very much. However, it is certain that they are important parameters to create models on the MICU dataset in classical ML algorithms.

In the COMBINE dataset, there is more than one parameter affecting the result label as in the SITARC and MICU datasets. In addition, the result label is classified into more than two categories. For these reasons, when determining the prediction approach, it may be useful to evaluate and compare algorithms in terms of gain, to produce more effective solutions in decision-making processes. In the proposed thesis, the COMBINED dataset consists of 8 attributes such as SITARC and MICU datasets, while the result label is divided into 5 different categories to push the limits of the algorithms.

Within the scope of the proposed thesis, the algorithms implemented into the COMBINED dataset were evaluated in terms of gain and loss. Because, in multi-class datasets such as the COMBINED dataset, the predicted value is in a class close to the true value is more acceptable than it is in a class far from the true value. The gain performance of the algorithms implemented in the COMBINED RDF dataset is given in Figure 6.13.

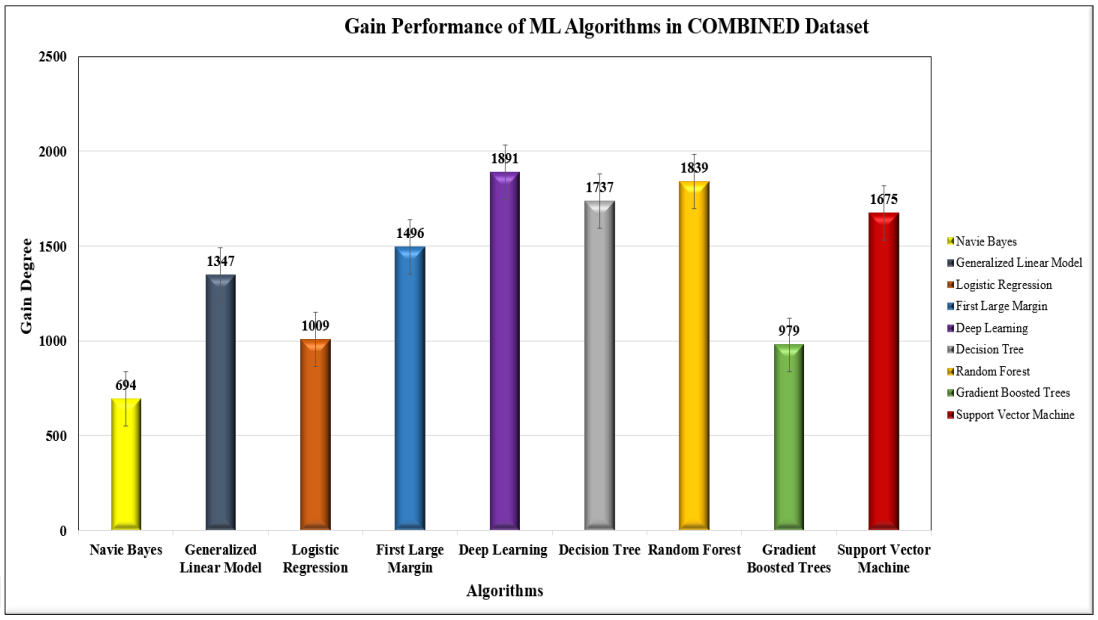


Figure 6.14 Comparison of gain of algorithms for the COMBINED dataset

A cost matrix was created to evaluate algorithms in terms of gain and loss. While creating this Cost matrix, the distance of the estimated value from the actual values was taken as a reference. The cost Matrix used for gain calculation in the COMBINED dataset is the same as the Cost matrix used in the SITARC and the MICU dataset. For this reason, how to create the Cost matrix will not be explained again in this section.

When the performance of the algorithms is examined in terms of gain, Figure 6.14 clearly are shown that the DL approach provides the best performance with 1,891 score as in the MICU dataset and in parallel with the accuracy performance. Following the DL algorithm, it is seen that RF approach with 1,839 score, DT approach with 1,737 score and SVM approach with 1,675 score. It is thought that these 4 algorithms can be used in terms of gain performance because they are higher than the average value for COMBINED dataset enriched with semantic web technologies.

Apart from these algorithms, the FLM approach with 1,496 points, the GLM approach with 1,347 score, and the LR approach with 1,009 score obtained. These results scores are close to the average gain score in the COMBINED dataset. However, it is obvious that these gain scores are insufficient for the prediction model required for a proactive system design. Following these approaches, it is certain that the GBT

algorithm with a score of 979 points cannot be used as a prediction approach in a proactive system design in the COMBINED dataset, by obtaining a score well below the average. Lastly, the NB algorithm obtained 694 scores while obtaining a poor result that could not be compared with other approaches.

When the performances of 9 classical ML algorithms implemented in the COMBINED RDF dataset are analyzed in terms of accuracy, Training Time, Scoring Time, Total Time, and Gain; unlike the SITARC and MICU datasets, there is no outstanding approach among implemented algorithms. In many respects, the approaches have shown similar results. However, when the experimental results are examined in terms of accuracy performance, it is understood that DL, DT, RF, and SVM algorithms stand out and can be used for prediction approach in a proactive system by looking at the above graphs and comments.

While some algorithm performances in the COMBINED dataset gave lower results in terms of accuracy compared to the SITARC dataset, they also gave better results compared to the MICU dataset. Although the datasets consist of similar attributes, these differences are thought to be due to the dataset characteristics. However, it should not be forgotten that although the SVM approach has provided good performance in terms of accuracy, the Total Time of this algorithm is very high compared to other algorithms, which may be insufficient for a system operated in real-time.

Likewise, the performance RF approach is insufficient, especially in terms of Scoring Time and Total Time, compared to the other two algorithms. Among the remaining two approaches, the DT algorithm provided better timing performance compared to the DL algorithm in terms of Total Time. However, the accuracy performances of these algorithms are slightly lower than DL. In the light of all these data, the advice of the project team is to use the DT algorithm with little compromise from accuracy in vital processes that need to perform instant analysis in the COMBINED RDF dataset, and where possible action plans need to be executed faster.

CHAPTER SEVEN
CONCLUSION AND FUTURE WORK

In the last decade, sensor-based systems have spread rapidly to all areas of daily life, especially in industrial areas, as a result of the become smaller so that they can be used in every system, developments in the academic environment, and the decrease in their prices. Such the widespread use of sensor systems has caused an enormous increase in sensor-based data, especially in internet environments. The representation, reusability, interpretation, and management of these large-scale sensor data on the Internet is still one of the areas that await effective solutions today.

Another difficulty with the sensor data is that the sensor data are generally heterogeneous in the data they produce due to different operating principles, different hardware, and different purposes. These heterogeneity detection methods may involve one or more of the small differences in operating systems, syntax, and data structure. In other words, the sensor data produced are generally specific to that system. Sensor data obtained within a specific system are not shared with other systems, they are not reused and it is very difficult to manage them in a common framework.

In addition to all these, the fact that sensor data is not encoded in a language that computers can understand makes sensor data difficult to understand and interpret by machines. Recent research in this area has focused on the joint representation and management of sensor data under a common roof. In this proposed thesis, a common framework has been created to enable machines to interpret sensor data on the Web by providing more advanced access and annotations. This system was named Ontology Framework for Heterogeneous Sensor Data (OF4HeS:Lite). The infrastructure of this common framework consists of the SSN.

In the proposed thesis, OF4HeS:Lite consists of roughly 3 different processing steps closely related to each other. Firstly, sensor nodes were created to collect determined environment variables. After that, IAQ data were collected in SITARC and MICU through these sensor nodes. In the second step, a common sensor ontology was created

by using the SSN framework in order to manage these sensor data collected from different environments and different platforms under the same roof. In this way, a common metadata and representation standard is provided for sensor data. Providing common metadata is an important layer to enable machines to better interpret sensor data. At the last stage of the proposed thesis, it has been tried to determine which ML approach is more effective on semantically enriched sensor data in order to create a proactive system design.

The performances of ML approaches may vary according to different situations, different samples, and different data sets. However, since the duration and difficulties of the training and testing stages may vary, they should be evaluated separately. When factors such as performance, scalability, flexibility, accuracy, and precision are considered, it is impossible to predict which method or algorithm is more suitable for a situation or data set.

To be able to say that an algorithm is the best for a case study or dataset, it must create a delicate balance especially in terms of accuracy, training time, test time, flexibility. Therefore, in the proposed thesis, careful experiments were conducted to determine the best estimation approach in every aspect while modeling for the future in RDF datasets encoded with Semantic metadata. For this reason, defining suitable prediction approaches for any data set requires a long and demanding experiment process that requires attention. The performances of the implemented approaches on 3 RDF data sets created within the scope of the thesis study are given below from bad to good.

A probabilistic based approach, NB is a supervised ML algorithm. This algorithm has been used in many studies as a prediction approach. In many studies, although this approach has given good results, up to which can be compared with approaches with more complex operating principles such as SVM and DL, it also remained far below the average in terms of gain and accuracy performance on the RDF datasets used in this study. Considering the time performances, it was observed that NB's time periods deteriorated with the increase in the number of samples, especially in the COMBINED

RDF data set. The experiments conducted in this study showed that the use of the NB approach in RDF data sets is not sufficient in terms of both accuracy performance and time performance.

Considering the performance of the LR algorithm in the data sets used in this thesis, it is obvious that it is one of the worst algorithms in terms of gain and accuracy performance, especially in the SITARC data set. It is a remote possibility to use this regression-based algorithm as a prediction approach in RDF data sets. Apart from this, in terms of time performance, while the LR algorithm achieves a performance close to the average value in SITARC and MICU data sets, it is seen that the timing performance in the COMBINED data set decreases significantly as the number of data increases, just like the NB algorithm. When considered in this respect, it seems impossible to use the LR approach as a prediction model.

Within the scope of the thesis, another approach that is implemented on RDF data hardnesses is the GBT algorithm which is the tree-based approach. While the GBT algorithm performed remarkably only in SITARC dataset in terms of gain and accuracy, it performed well below average in other datasets. In total, the gain accuracy performance in the 3 data sets was below average. Although this algorithm performs better than the average in all 3 data sets in terms of Total Time, which is the most important indicator in terms of timing in the Prediction model, it is not possible to use it for the RDF data sets used in the study due to the insufficient accuracy rate.

According to the experimental results, the RF algorithm, which is another tree-based approach, obtained an average value in terms of gain and accuracy in total. Although this approach has managed to be the best algorithm in terms of gain and accuracy in the SITARC dataset, its performance has significantly decreased in the COMBINED dataset, where the number of samples increased, and the MICU dataset where the clarity between the labels decreased. Although it has performed better than the average value in all three data sets in terms of timing, it is thought that it is a remote possibility to use it as a prediction model when all the data sets created within the scope of the thesis study are considered when compared with other algorithms.

When the results of the algorithms implemented in all three data sets are examined, it is seen that the SVM algorithm, which is frequently used as the prediction approach, is above average in terms of gain and accuracy performance. However, it is obvious that the SVM algorithm performs very poorly in terms of Training Time, Scoring Time, and Total Time. Therefore, it is thought that the SVM approach should not be used in semantic web technologies in semantically enriched SITARC, MICU, and COMBINED datasets, especially in proactive systems where instant decisions need to be taken.

When the results of the implemented algorithms are examined in all three data sets, as the prediction approach, the FLM algorithm has a similar performance with the DT algorithm in terms of gain and accuracy performance. Considering accuracy and gain performance, it is considered to be a convenient approach for RDF datasets. However, considering the average of the timing performance in 3 data sets, it could not provide the performance shown by many algorithms. Therefore, it is thought that this approach cannot be used as a prediction approach in RDF data sets, while there are approaches that show the same accuracy and provide better timing performance.

Considering the gain and accuracy performances, another approach that performs well is the GLM algorithm, which shows a close performance to DT and FLM approaches. Although not the best approach, experiments have proven that the GLM algorithm, which performs close to the best approach in all 3 data sets, can be used for RDF data sets in terms of accuracy and performance. When evaluated in terms of time performances, it has achieved a good performance according to the FLM approach, which is one of the algorithms providing similar accuracy, but poor performance according to the DT approach. For this reason, it is predicted that the DT approach with similar accuracy rates has a better time performance, while the GLM algorithm cannot be used.

The last tree-based approach implemented to the created data sets is the DT algorithm. This approach provided approximately 70% accuracy performance and a

parallel gain performance when averaged over 3 data sets where it was implemented in terms of accuracy performance. Considering this gain and accuracy performance it demonstrates, it is thought that it can be used for RDF data sets. Considering the performance of the DT algorithm in terms of time criteria, it showed an average performance in terms of Training Time in all 3 data sets used and showed the best performance in 3 data sets in terms of Scoring Time and Total Time. According to the results of the Experiment, it is thought that the DT algorithm can be used for RDF data sets in terms of both accuracy performance and time performance.

Finally, considering the average accuracy performances of the 3 data sets used within the scope of the thesis study, the best approach with a value of approximately 80% has definitely been the DL algorithm, which has been frequently used as the prediction approach recently. However, due to the complexity of the working principle of the DL approach, the time performances are often below average. As a result, it is thought that it is appropriate to use the DL approach, which has high accuracy but low time performance in data sets where the number of samples in the data set is low and the need for frequent modeling is not felt. However, it may be more appropriate not to use the DT approach to ensure the timely response of the system in large data sets, which are frequently modeled. In this case, it may be more appropriate to use faster responsive approaches such as DT and GLM, with some compromise on accuracy.

"OF4HeS:Lite" proposed in this study is a low-level sensor ontology that provides a better interpretation of sensor data. It is thought that OF4HeS:Lite will guide mid-level and high-level ontologies planned to be done next. This model proposed in future studies can be combined with different domains, different platforms, and different systems to expand its scope. With this extended model, sensor data can be used to make a common inference.

Although the proposed sensor ontology associates the data semantically, the complexity of the semantic techniques often causes the processing times to increase. A new model can be created that includes minimum concepts to ensure that the proposed semantic systems respond in a reasonable time acceptable to data consumers.

Object properties and data properties can be used within the scope of the minimum concept. Thus, the number of triples in the RDF database may be reduced and the system can be more efficient.

REFERENCES

Abidin, N. Z., Ismail, A. R., & Emran, N. A. (2018). Performance analysis of machine learning algorithms for missing value imputation. *International Journal of Advanced Computer Science and Applications, 9* (6), 442–447.

Adeleke, J., Moodley, D., Rens, G., & Adewumi, A. (2017). Integrating statistical machine learning in a semantic sensor Web for proactive monitoring and control. *Sensors, 17* (4), 807-829.

Agrawal, R., Almaden, T., & Swami, A. (1993). Mining association in large databases. *International Conference on Management of Data*, 207–216.

Akpınar, H. (2000). Veri tabanlarında bilgi keşfi. *İstanbul Üniversitesi İşletme Fakültesi Dergisi, 29* (1), 1–22.

Aktaş, Ö., Milli, M., Lakestani, S., & Milli, M. (2020). Modelling sensor ontology with the sosa/ssn frameworks: A case study for laboratory parameters. *Turkish Journal of Electrical Engineering & Computer Sciences, 28* (5), 2566–2585.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Journal of Computer Networks, 38* (4), 393–422.

Aliyev, N. (2019). *Kablosuz algılayıcı alarda en kısa yol algortimalarının incelenmesi.* MSc Thesis, Dokuz Eylül University, İzmir.

Altay, O., & Ulaş, M. (2018). Anlamsal web kullanılarak ilaç ontolojisi çıkarılması. *Fırat University Journal of Engineering Science, 30* (1), 169–174.

Ams, C. (2017). *Ultra-low power digital gas sensor for monitoring indoor air quality*. Retrieved October 14, 2020 from http://ams.com/eng/Products/Environmental-sensors/air-quality-sensors/ccs811

Apache, J. (2011). *Apache Software Foundation.* Retrieved October 12, 2019, from https://jena.apache.org/documentation/fuseki2/index.html

Apollo. (n.d.). *Ontology building editor developed by Knowledge Media Institute*. Retrieved October 26, 2020, from http://apollo.open.ac.uk/

Arduino, H. (n.d.). *Arduino uno R3.* Retrieved October 17, 2020, from https://www.arduino.cc/en/main/arduinoboardunosmd

Avancha S, Patel C, Joshi A. Ontology-driven adaptive sensor networks. In: *Proceedings of 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services; Boston, USA, 2004*, 194-202.

Aydemir, B. (2017). *Veri Madenciliği yöntemleri kullanarak meslek yüksek okulu öğrencilerinin akademik başarı tahmini.* MSc Thesis, Pamukkale University, Denizli.

Aydemir, E. (2018). *Weka ile Yapay Zeka* (1st ed.). İstanbul: Seckin Yayınevi.

Balmin, A., & Papakonstantinou, Y. (2005). Storing and querying XML data using denormalized relational databases. *The International Journal on Very Large Data Bases*, *14* (1), 30–49.

Barbur, G., Blaga, B., & Groza, A. (2011). OntoRich: A support tool for semi-automatic ontology enrichment and evaluation. *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing, ICCP 2011*, 129–132.

Barnaghi, P., & Presser, M. (2010). Publishing linked sensor data. *CEUR Workshop Proceedings*, 668-673.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd ed.). Chichester : The Wiley Network Publishing.

Barragán, H. (2004). *Wiring : prototyping physical interaction design. Interaction Design Institute, Ivrea, Italy.* Phd Thesis, Interaction Design Institute Ivrea, Italy.

Baxter, R., Hastings, N., Law, A., & Glass, E. J. . (2008). Arduino uno R3. *Animal Genetics*, *39* (5), 561–563.

Berendt, B., Hotho, A., & Stumme, G. (2002). Towards semantic Web mining. *The International Semantic Web Conference*, 264–278.

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., & Taylor, K. (2017). IoT-Lite: A lightweight semantic model for the internet of things and its use with dynamic semantics. *Personal and Ubiquitous Computing*, *21* (3), 475–487.

Berners-lee, T., Hendler, J., & Lassila, O. (2001). The semantic Web. *Scientific American*, *284* (5), 35–43.

Bhandari, S., Bergmann, N., Jurdak, R., & Kusy, B. (2017). Time series data analysis of wireless sensor network measurements of temperature. *Journal of Sensors*, *17* (6), 1221.

Bilgin, T., & Gökhan, A. C. U. N. Yazılım Hata Logları Kullanılarak Veri Madenciliği Uygulaması Gerçekleştirilmesi. *Marmara Fen Bilimleri Dergisi*, *27* (1), 14-20

Borst, W. N. (1997). *Construction of engineering ontologies for knowledge sharing and reuse*. Phd Thesis, University of Twente, Netherlands.

Boubiche, S., Boubiche, D. E., & Toral-cruz, H. (2018). Big data challenges and data aggregation strategies in wireless sensor networks. *IEEE Access*, *6*, 20558–20571.

Boubrima, A., Bechkit, W., & Rivano, H. (2017). Optimal WSN deployment models for air pollution monitoring. *IEEE Transactions on Wireless Communications, 16* (5), 2723–2735.

Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., et al. (2011). New generation sensor Web enablement. *Journal of Sensors, 11* (3), 2652–2699.

Brown, S. (2008). *High quality indoor environments for sustainable office buildings. Grey Article* (1st ed.). Singapore: Springer.

Büchner, A. G., Anand, S. S., & Hughes, J. G. (2014). Data mining in manufacturing environments : goals , techniques and applications. *Studies in Informatics and Control, 1* (1), 1–8.

Calbimonte, J., Jeung, H., Corcho, O., & Aberer, K. (2012). Enabling query technologies for the semantic sensor Web. *International Journal on Semantic Web & Information Systems, 8* (1), 43–63.

Camastra, F., & Vinciarelli, A. (2015). *Machine learning for audio, image and video Analysis* (1st ed.). London: Springer.

Ceyhan, E. B., & Sağiroğlu, Ş. (2013). Security issues on wireless sensor networks and the possible precautions. *Journal of Polytechnic, 16* (4), 155–163.

Cheffena, M. (2012). Industrial wireless sensor networks: channel modeling and performance evaluation. *Journal on Wireless Communications and Networking, 297* (1), 12–20.

Chouikhi, S., El, I., Ghamri-doudane, Y., & Azouz, L. (2015). A survey on fault tolerance in small and large scale wireless sensor networks. *Computer Communications, 69 (1),* 22–37.

Chowdhury, K. (2017). *Mastering Visual Studio 2017*. Birmingham: Packt Publishing Ltd.

Co, N. F. (2015). *Nova laser pm sensor specification.* Retrieved September 27, 2020, from https://cdn-reichelt.de/documents/datenblatt/x200/sds011-datasheet.pdf

Coenen, F. (2011). Data Mining: Past, present and future. *Knowledge Engineering Review*, *26* (1), 25–29.

Compton, M., Barnaghi, P., Bermudez, L., Castrod, R. G., Corcho, O., Cox, S., et al. (2011). The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web*, *17* (1), 25–32.

Darryl, N. D., & Rahman, M. M. (2016). Missing value imputation using stratified supervised learning for cardiovascular data. *Journal of Informatics and Data Mining, 1 (2),* 1-13.

Das, S. N., Misra, S., Member, S., Wolfinger, B. E., & Obaidat, M. S. (2016). Temporal-correlation-aware dynamic self-management of wireless sensor networks. *IEEE Transactions on Industrial Informatics*, *12* (6), 2127–2138.

Deb, R., & Liew, A. W. C. (2016). Missing value imputation for the analysis of incomplete traffic accident data. *Journal Information Sciences*, *339* (1), 274–289.

Decker, S., Harmelen, F. Van, Broekstra, J., Erdmann, M., Fensel, D., Horrocks, I., et al. (2000). The semantic Web: On the respective roles of XML and RDF. *IEEE Internet Computing*, *4* (10), 63–74.

DotNetRDF. (2020). *An open source .NET Library for RDF*. Retrieved October 22, 2020, from https://www.dotnetrdf.org/

Ekinci Eser, E. (2006). *Bir ontolji eşleştirme aracı gerçekleştirimi*. Msc Thesis, Ege University, İzmir.

Electronics, H. H. (2018). *Mq-7 carbon monoxide datasheet.* Retrieved September 29, 2019, from https://www.pololu.com/file/0J313/mq7.pdf%0A%0A

Elsayed, W., Elhoseny, M., Sabbeh, S., & Riad, A. (2018). Self-maintenance model for wireless sensor networks R. *Computers and Electrical Engineering*, *70* (2018), 799–812.

Engel, A., & Koch, A. (2016). Heterogeneous wireless sensor nodes that target the internet of things this article summarizes the architectural design decisions of the hardware. *IEEE Journal of Micro*, *36* (6), 8–15.

Ertugrul, I., Organ, A., & Savli, A. (2013). The determination of patient profile at Pamukkale University as relater to the application of data mining. *Pamukkale University Journal of Engineering Sciences*, *19* (2), 97–103.

Feng, C., Li, J., Sun, W., Zhang, Y., & Wang, Q. (2016). Impact of ambient fine particulate matter exposure on the risk of influenza-like-illness: A time-series analysis in Beijing, China. *Environmental Health: A Global Access Science Source*, *15* (1), 1–12.

Ganesh, S. (2002). Data mining. The 6th International Conference on Teaching Statistics (ICOTS 6), 1–4.

Gemici, B. (2012). *Veri Madenciliği ve bir uygulaması*. Msc Thesis Dokuz Eylül University, İzmir.

Geylani, M. (2018). Kablosuz algılayıcı ağlarda melez çoğullama yöntemleri kullanılarak zaman bağımsız yeni bir iletim tekniğinin geliştirilmesi. MSc Thesis, Bitlis Eren University, Bitlis.

Goodwin, C., & Russomanno, D. J. (2006, April). An ontology-based sensor network prototype environment. In *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*, 1-2.

Graph, A. (n.d.). *AllegroGraph is a modern, high-performance, persistent graph database*. Retrieved October 14, 2019, from https://franz.com/agraph/allegrograph/

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, *43* (5), 907–928.

Gungor, V. C., Hancke, G. P., & Member, S. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, *56* (10), 4258–4265.

Haller, A., Janowicz, K., Cox, S. J. D., Lefrançois, M., Taylor, K., Le Phuoc, D., et al. (2018). The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Journal of Semantic Web*, *10* (1), 9–32.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques* (3 rd.). USA: Elsevier.

Harrington, P. (2012). *Machine learning in action* (First). Shelter Island: Manning Publications.

Hebeler, J., Fisher, M., Blace, R., & Perez-Lopez, A. (2009). *Semantic Web programming*. Indianapolis, Indiana.: Wiley Publishing, Inc.

Henson, C., Neuhaus, H., & Sheth, A. (2009). An ontological representation of time series observations on the semantic sensor Web. *The Ohio Center Of Excellence In Knowledge-Enabled Computing*, *6* (1), 1–15.

Hu, A., Wang, H., & Wan, J. (2013). Design of WSN based remote monitoring system for environmental parameters in substation. *International Journal of Smart Grid and Clean Energy Design*, *2* (1), 1–6.

Huang, V., & Javed, M. K. (2008). Semantic sensor information description and processing. *2nd International Conference of Sensor Technology Applications, 15* (2), 456–461.

Hussain, F., Cebi, Y., & Shah, G. (2008). A multievent congestion control protocol for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, *2008* (1), 803271.

Hussin, M., Ismail, M. R., & Ahmad, M. S. (2017). Air-conditioned university laboratories: comparing CO_2 measurement for centralized and split-unit systems. *Journal of King Saud University - Engineering Sciences*, *29* (2), 191–201.

Ibrahim, M., & Moselhi, O. (2015). Self-calibrated WSN for indoor trackıng and control of construction operations. *5th International/11th Construction Specialty Conference*, *15* (2), 1–10.

Jain, A. K., Murty, M. N., & P. J. F. (1999). Data clustering: a review. *Association for Computing Machinery Computer Survey*, *31* (3), 264–323.

Janowicz, K., Bröring, A., Stasch, C., & Everding, T. (2010). Towards meaningful URIs for linked sensor data. *CEUR Workshop Proceedings*, 640.

Janowicz, K., & Compton, M. (2010). The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. *CEUR Workshop Proceedings*, 668.

Janssen, N. A. H., Fischer, P., Marra, M., Ameling, C., & Cassee, F. R. (2013). Short-term effects of pm2.5, pm10 and pm2.5-10 on daily mortality in the Netherlands. *Science of the Total Environment*, *463* (3), 20–26.

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B. C., & Hendler, J. (2006). Swoop: A web ontology editing browser. *Journal of Web Semantics*, *4* (2), 144–153.

Karim, F., & Zeadally, S. (2016). Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews*, *55* (1), 1041–1054.

Kaur, M., & Garg, P. (2016). Improved distributed fault tolerant clustering algorithm for fault tolerance in WSN. *2016 International Conference on Micro-Electronics and Telecommunication Engineering*, 35–41.

Kaur, S., & Mir, R. N. (2015). Quality of service in WSN-a review. *International Journal of Computer Applications*, *113* (18), 42–46.

Knime. (n.d.). *End to end data science.* Retrieved October 26, 2020, from https://www.knime.com/

Koyuncugil, A. S., & Özgülba, N. (2009). Data mining: using and applications in medicine and healthcare. *Journal of Information Technologies*, *2* (2), 21–32.

Krawczyk, D. A., Rodero, A., Gładyszewska-Fiedoruk, K., & Gajewski, A. (2016). CO_2 concentration in naturally ventilated classrooms located in different climates measurements and simulations. *Energy and Buildings*, *129* (1), 491–498.

Krzyzanowski, M., & Cohen, A. (2008). Update of WHO air quality guidelines. *Air Quality, Atmosphere & Health*, *1* (1), 7–13.

Kuster, C., Hippolyte, J., & Rezgui, Y. (2020). Advances in engineering software The UDSA ontology: An ontology to support real time urban sustainability assessment. *Advances in Engineering Software*, *140* (2019), 102731.

Le-Phuoc, D., & Hauswirth, M. (2009). Linked open data in sensor data mashups. *CEUR Workshop Proceedings*, *522*, 1–16.

Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., et al.. (2017). *Semantic sensor network XG final report*. Retrieved October 10, 2020, from https://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/

Liu, J., Li, Y., Tian, X., Sangaiah, A. K., & Wang, J. (2019). Towards semantic sensor data: An ontology approach. *Sensors (Switzerland)*, *19* (5), 1–21.

Liu, T., & Manager, B. (n.d.). *Dht 22 temperature and humidity sernsor datasheet.* Retrieved September 25, 2019, from https://www.sparkfun.com/datasheets/Sensors/Temperature/dht22.pdf

Mansour, E., Chbeir, R., & Arnould, P. (2019). HSSN : An ontology for hybrid semantic sensor networks. *International Database Applications*, *4* (6), 1–8.

Microcontroller, A. V. R. (n.d.). Atmega328P datasheet, 1–294.

Miller, E. (2001). *Digital libraries and semantic web layers.* Retrieved September 24, 2020, from https://www.w3.org/2001/09/06-ecdl/slide17-0.html

Mishra, S. K., & Singh, V. K. (2016). Ontology development for software trackıng informatıon system. *International Journal of Research in Engineering and Technology*, *5* (3), 138–140.

Musen, M. A., & Team, P. (2015). The protégé project. *AI Matters*, *1* (4), 4–12.

MySensors Library. (n.d.). *Home automation and internet of things.* Retrieved October 24, 2020, from https://www.mysensors.org/

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., et al. (1991). Enabling technology for knowledge sharing. *AI Magazine, 12* (3), 36–56.

Nordic Semiconductor [NS]. (2008). nRF24L01+ Single Chip 2.4GHz Transceiver datasheet.

Onal, A. C., Berat Sezer, O., Ozbayoglu, M., & Dogdu, E. (2017). Weather data analysis and sensor fault detection using an extended IoT framework with semantics, big data, and machine learning. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, *2018-January*, 2037–2046.

Othman, A., & Maga, D. (2018). Relation between security and energy consumption in wireless sensor network. In *2018 New Trends in Signal Processing*. Demanovska Dolina, Slovakia: IEEE.

Övünç, Ö. (2004). *Anlamsal web için bir ontoloji ortamı tasarımı ve gerçekleştirimi.* MSc Thesis, Ege University, İzmir.

Özdağ, R. (2016). The Solution of the k-coverage problem in wireless sensor networks. In *24th Signal Processing and Communications Applications Conference* Zonguldak, Turkey, 873–876.

Patni, H. K., Henson, C. A., & Sheth, A. P. (2010). Linked sensor data. In *International Symposium on Collaborative Technologies and Systems* 362–370.

Percivall, G., Reed, C., & Davidson, J. (2007). *Sensor Web enablement: overview and high level architecture* . Berlin: Springer.

Pighix. (2013). *Arduino R3 and Atmega328 pinout diagram*. Retrieved October 20, 2020, from www.pighixxx.com

Powel, J., & Hopkins, M. (2015). *A Librarian's guideto graphs, data and the semantic* Web. USA: Elsevier.

Punnoose, R., Crainiceanu, A., & Rapp, D. (2012). Rya: a scalable RDF triple store for the clouds. In *Proceedings of the 1st International Workshop on Cloud Intelligence* 32–40.

Qiu, H., Yu, I. T. S., Tian, L., Wang, X., Tse, L. A., Tam, W., et al. (2012). Effects of coarse particulate matter on emergency hospital admissions for respiratory diseases: A time-series analysis in Hong Kong. *Environmental Health Perspectives*, *120* (4), 572–576.

Qiu, T., Member, S., Zhao, A., & Member, S. (2017). ROSE:robustness strategy for scale-free. *IEEE/ACM Transactions on Networking*, *25* (5), 2944–2959.

Radhika, S., & Rangarajan, P. (2019). On improving the lifespan of wireless sensor networks with fuzzy based clustering and machine learning based data reduction. *Applied Soft Computing Journal*, *83* (1), 105610.

Rani, A., & Kumar, S. (2017). A Survey of security in wireless sensor networks. In *3rd IEEE International Conference on Computational Intelligence and Communication Technology (IEEE-CICT), Ghaziabad, India,* 3–7.

Ratna, A., & Hansdah, R. C. (2015). Ad Hoc networks a model for the classification and survey of clock synchronization protocols in WSNs. *Ad Hoc Networks*, *27* (1), 219–241.

Salle, A., Idiart, M., & Villavicencio, A. (2016). Enhancing the lexvec distributed word representation model using positional contexts and external memory alexandre. *ArXiv preprint arXiv:1606.01283* (2016), 1-6.

Scale-free, M. C., Qiu, T., Member, S., Liu, J., Si, W., & Wu, D. O. (2019). Robustness optimization scheme with wireless sensor networks. *IEEE/ACM Transactions on Networking*, *27* (3), 1028–1042.

Segaran, T., Evans, C., & Taylor, J. (2009). *Programming the semantic Web*. *Semantic Web services processes and applications* (2nd ed.). USA: O'Reilly Media Inc.

Sezer, O. B., Dogdu, E., & Ozbayoglu, A. M. (2018). Context-aware computing, learning, and big data in internet of things: A Survey. *IEEE Internet of Things Journal*, *5* (1), 1–27.

Sharma, S., Bansal, S., & Bansal, R. K. (2013). Issues and challenges in wireless sensor networks. *2013 International Conference on Machine Intelligence and Research Advancement 2013 International Conference on Machine Intelligence Research and Advancement Issues,* 58–62.

Sharmin, M., Raij, A., Epstien, D., Nahum-Shani, I., Beck, J. G., Vhaduri, S., et al. (2015). Visualization of time-series sensor data to inform the design of just-in-time adaptive stress interventions. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*, 505–516.

Sheth, A. (1999). Changing focus on interoperability in information systems: from systems, syntax, structure to semantics. *Interoperating Geographic Information Systems*, *1* (1), 248-259.

Sheth, A., Henson, C., & Sahoo, S. S. (2008). Semantic sensor Web. *IEEE Internet Computing*, *12* (4), 78–83.

Stanford Team. (2016). *A free, open-source ontology editor.* Retrieved October 26, 2020, from https://protege.stanford.edu/

Stanke, D. A., Danks, R. A., Muller, C. O., Hedrick, R. L., Fisher, F. J., Osborn, J. E., et al. (2007). Ventilation for acceptable indoor air quality. *Ashrae Standard*, *2007* (62.1-2007), 1–41.

Strandberg-larsen, A. K., Grønbæk, M., Andersen, A. N., Strandberg-larsen, K., Gronbcek, M., Andersen, A. N., & Andersen, P. K. (2016). Alcohol drinking pattern during pregnancy and risk of infant mortality a human teratogen. *Journal of Epidemiology*, *20* (6), 884–891.

Sunrom Technologies. (2008). *Light dependent resistor (ldr) Data Sheet. Sunrom Technology.* Retrieved October 20, 2020, from http://igem.org/wiki/images/1/1a/file-t--technion_israel-hardwarespecsldr.pdf

Tableau. (n.d.). *Analyze data with intuitive drag & drop products.* Retrieved October 26, 2020, from https://www.tableau.com/

Toma, D. M., Mart, E., Reilly, T. C. O., Delory, E., Pearlman, J. S., Fellow, L., et al. (2018). A sensor Web architecture for integrating smart oceanographic sensors into the Semantic Sensor Web. *IEEE Journal of Oceanic Engineering 43* (4), 830–842.

Tsai, C., Tsai, P., Pan, J., & Chao, H. (2015). Microprocessors and microsystems metaheuristics for the deployment problem of WSN : a review. *Microprocessors and Microsystems*, *39* (8), 1305–1317.

Tubaishat, M., & Madria, S. (2003). Sensor networks: an overview. *IEEE Potentials*, *22* (2), 20–23.

Vahaplar, A., & İnceoğlu, M. M. (2001). Veri madenciliği ve elektronik ticaret. *Türkiye'de Internet Konferansları VII*, 1-8.

Virtuoso, (n.d.). *Data-driven agility without compromise. (n.d.).* Retrieved October 16, 2019, from https://virtuoso.openlinksw.com/

Wang, C., Chen, N., Wang, W., & Chen, Z. (2018). A hydrological sensor web ontology based on the SSN ontology: A case study for a flood. *ISPRS International Journal of Geo-Information*, *7* (1), 2.

Wang, F., Hu, L., Zhou, J., Hu, J., & Zhao, K. (2017). A semantics-based approach to multi-source heterogeneous information fusion in the internet of things. *Soft Computing*, *21* (8), 2005–2013.

Wang, Q. (2010). Wireless sensor networks. In *An Introduction* 1–17.

Wang, R.-Q., & Kong, F.-S. (2007). Semantic-enhanced personalized recommender system. *Sixth International Conference on Machine Learning and Cybernetics*, *5* (3), 19–22.

Weiten, M. (2009). OntoStudio as a ontology engineering environment. In *Semantic Knowledge Management* 51–60. Berlin, Heidelberg: Springer.

Weka. (n.d.). *The workbench for machine learning.* Retrieved October 26, 2020, from https://www.cs.waikato.ac.nz/ml/weka/

Wenquan, J., & Kim, D. H. (2018). Design and implementation of e-health system based on semantic sensor network using IETF YANG. *Journal of Sensors*, *18* (2), 629–654.

Westphal, M., Ylä-Herttuala, S., Martin, J., Warnke, P., Menei, P., Eckland, D., et al. (2013). Adenovirus-mediated gene therapy with sitimagene ceradenovec followed by intravenous ganciclovir for patients with operable high-grade glioma. *Journal of The Lancet Oncology*, *14* (9), 823–833.

Wilbur, S., Williams, M., Williams, R., Scinicariello, F., Klotzbach, J. M., Diamond, G. L., & Citra, M. (2012). *Toxicological profile for carbon monoxide*. *U.S. Agency for Toxic Substances and Disease Registry*. (1st ed.). Atlanta: SRC, Inc.

World Health Organization [WHO]. (2010). *World Health Organization guidelines for air quality: selected pollutants. Indian pediatrics*. Retrieved October 12, 2020, from https://www.euro.who.int/__data/assets/pdf_file/0009/128169/e94535.pdf

World Wide Web Consortium [W3C] (2017). Retrieved October 24, 2020, from https://www.w3.org/TR/vocab-ssn/

Yang, J. H., Cheng, C. H., & Chan, C. P. (2017). A time-series water level forecasting model based on imputation and variable selection method. *Computational Intelligence and Neuroscience, 2017* (1), 128-135.

Yang, S., & Byun, H. (2020). Biologically inspired reasoning scheme for semantic sensor network ontology in efficient disaster surveillance. *Journal of Sensors and Materials*, *32* (6), 2237–2245.

Yılmaz, S. (2009). *Yaşam boyu müşteri değeri modellemesi üzerine bir örnek uygulaması*. MSc Thesis, Yıldız Teknik University, İstanbul.

Zahangeer A. M., Armin, E., Haque, M. M., Halsey, J., & Qayum, M. A. (2018). Air pollutants and their possible health effects at different locations in Dhaka City. *Journal of Current Chemical and Pharmaceutical Sciences, 8* (1), 110-120.

Zhao, F., Guibas, L. J., & Guibas, L. (2004). *Wireless sensor networks: An Information Processing Approach.* (1st ed.). London: Elsevier Publishing.

Zhong, N., & Zhou, L. (1999). Methodologies for knowledge discovery and data mining. *Third Pacific-Asia Conference*, 532-540.

Zhu, X., Zhang, P., Lin, X., Shi, Y., Science, C., & Raton, B. (2007). Active learning from data streams. *Seventh IEEE International Conference on Data Mining*, *12* (3), 757–762.

Zion, D. O., & Messer, H. (2014). Envelope only TDOA estimation for sensor network. *IEEE 8th Sensor Array and Multichannel Signal Processing Workshop Envelope*, *3* (1), 229–232.