



POLITECNICO DI MILANO

Master in

Building Information Modelling



European Master in  
Building Information Modelling

BIM, IOT and BMS for the management of buildings and  
infrastructures. The digital twin for sustainable development

Supervisor:

Alessandro Zichi

Author:

Mohammad Hassan Meshkini



Co-funded by the  
Erasmus+ Programme  
of the European Union

a.a. 2020/2021

## **AUTHORSHIP RIGHTS AND CONDITIONS OF USE OF THE WORK BY THIRD PARTIES**

This is an academic work that can be used by third parties, as long as internationally accepted rules and good practices are respected, particularly in what concerns to author rights and related matters.

Therefore, the present work may be used according to the terms of the license shown below.

If the user needs permission to make use of this work in conditions that are not part of the licensing mentioned below, he/she should contact the author through the BIM A+ Secretariat of Politecnico di Milano.

*License granted to the users of this work*



**Attribution  
CC BY**

<https://creativecommons.org/licenses/by/4.0/>

## ACKNOWLEDGEMENTS

This dissertation wouldn't have been possible without the support of the Erasmus Mundus scholarship and BIM A+ consortium that allowed me to develop my understanding regarding BIM and insights in the following project.

I owe an enormous debt of gratitude to my family who gave me unconditional support and encouraged me to push the boundaries in order to advance to the fullest.

I'm also immensely grateful to all friends and professionals who assisted me to think outside the box and provide me the pleasure to practice their ideas, suggestions, and constructive criticisms.

## **STATEMENT OF INTEGRITY**

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of ethics and conduct of Politecnico di Milano.

## SOMMARIO

Title: BIM, IOT e BMS per la gestione di edifici e infrastrutture. Il gemello digitale per lo sviluppo sostenibile

L'evoluzione dell'applicazione BIM è il Facility management per la digitalizzazione dell'Attività Edilizia. L'applicazione BIM è diventata l'occasione per aumentare la salvaguardia della città durante il suo intero ciclo di vita, partendo dalla conoscenza dell'effettiva conservazione dell'edificio e delle prospettive di utilizzo futuro.

Partendo dall'analisi dell'evoluzione del sistema IOT Control, del CDE e dell'evoluzione del browser dell'applicazione, l'obiettivo è creare un processo per simulare la costruzione di attività nelle città e nelle infrastrutture. L'applicazione sarà orientata a diversi casi di studio utilizzando diversi software (es. Autodesk, Bentley). Vengono inoltre affrontate le problematiche relative alla progettazione dei diversi sistemi che compongono la tecnologia CIB (sistemi di controllo e gestione dei sistemi, sicurezza, voce-dati-immagini e comunicazione informatica) e la metodologia più appropriata per raggiungere una buona capacità di gestione intelligente.

Partendo da GIS e rilievi ottenuti con diverse metodologie, lo studente svilupperà la simulazione di città e infrastrutture, compresa l'attività di costruzione. Lo scopo finale dell'FM sarà definito per l'intero ciclo di vita delle città e delle infrastrutture con riferimento a casi studio.

**Parole chiave:** BIM, BMS, gestione delle strutture, IoT

## ABSTRACT

The evolution of BIM application is the Facility management for the digitalization of the Activity Construction. The BIM application became the opportunity to increase the safeguard of the city during its entire life cycle, starting from the knowledge of the actual conservation of the building and the prospective for the future use.

Starting from the analysis of the evolution of the IOT Control system, the CDE and the evolution of the application browser, the focus is to create a process to simulate activity construction in cities and infrastructures. The application will be oriented to different case studies using different software (e.g. Autodesk, Bentley). Issues related to the design of different systems making up the CIB technology (systems of control and management of systems, security, voice-data-images and computer science communication) and the most appropriate methodology to achieve a good intelligent management capacity are also addressed.

Starting from GIS and survey obtained with different methods, the student will develop the simulation of cities and infrastructures, including the activity construction too. The final purpose of the FM will be defined for the whole life-cycle of cities and infrastructures with reference to case studies.

**Keywords:** BIM, BMS, facility management, IOT

# TABLE OF CONTENTS

1. INTRODUCTION.....	13
2. FACILITY MANAGEMENT.....	15
2.1. INTRODUCTION.....	15
2.2. INDOOR ENVIRONMENTAL QUALITY .....	16
2.3. COMPUTERIZED MAINTENANCE MANAGEMENT SYSTEM .....	16
2.4. BUILDING AUTOMATION SYSTEM (BAS) OR BUILDING MANAGEMENT SYSTEM (BMS) .....	18
2.5. SMART BUILDING AND SMART CITY .....	18
3. BUILDING INFORMATION MODELING (BIM).....	21
3.1. INTRODUCTIONS .....	21
3.2. BIM AND FACILITY MANAGEMENT .....	23
3.3. LOD.....	24
3.4. INTEROPERABILITY .....	26
3.5. CDE.....	26
4. DIGITAL TWIN.....	31
4.1. INTRODUCTION.....	31
4.2. INDUSTRY 4.....	32
4.3. DIGITAL TWIN ECOSYSTEM AND PLATFORMS .....	35
5. INTERNET OF THINGS (IOT).....	37
5.1. INTRODUCTION.....	37
5.2. IOT ECOSYSTEM AND PROTOCOLS.....	38
5.2.1. Perception layer.....	40
5.2.2. Network layer.....	42
5.2.3. Application layer.....	43
5.3. IOT DEVICES .....	44
6. CLOUD COMPUTING .....	49
6.1. INTRODUCTION.....	49
6.2. IAAS .....	50
6.3. PAAS.....	51
6.4. SAAS.....	51
6.5. FOG AND EDGE COMPUTING.....	52
7. CASE STUDY .....	57
7.1.....	INTRODUCTION
57	
7.2. METHODOLOGY.....	59
7.3. COLLECTING THE ENVIRONMENTAL DATA AND UPDATING THE BIM MODEL .....	62
7.3.1. Creating the BIM model and defining the rooms.....	62

7.3.2.	Raspberry Pi (Hardware and Software).....	63
7.3.3.	Connecting the light sensor .....	65
7.3.4.	Reading telemetry data from the sensor and sending them to the Google Sheets via Raspberry Pi .....	66
7.3.5.	Reading telemetry data from Google Sheets and storing it in a local CSV file.....	67
7.3.6.	Importing the telemetry data into the Autodesk Revit and updating the BIM model....	67
7.4.	GETTING THE COMMANDS FROM THE BIM MODEL AND SENDING THEM TO THE MOTOR.....	69
7.4.1.	Creating the parametric family of the window .....	69
7.4.2.	Connecting the Motor .....	70
7.4.3.	Defining scenarios for opening and closing the window’s louvers .....	72
7.4.4.	Modifying the model based on the scenarios, getting data from Autodesk Revit, and storing it in a local CSV file .....	74
7.4.5.	Sending the telemetry data from CSV file to the Google Sheets.....	75
7.4.6.	Reading the data from Google Sheets and sending it to the motor.....	75
8.	CONCLUSIONS .....	77
	REFERENCES.....	79
	LIST OF ACRONYMS AND ABBREVIATIONS.....	87
	APPENDICES .....	89
	APPENDIX 1: VNC VIEWER .....	90
	APPENDIX 2: SCRIPTS .....	92

## LIST OF FIGURES

Figure 1 - Simplified CMMS module structure .....	17
Figure 2 - Factors in the framework of Smart City .....	19
Figure 3 - Smart City Components .....	20
Figure 4 - Key Technologies in Smart Buildings.....	20
Figure 5 - BIM uses.....	22
Figure 6 - life cycle cost budget division of a building.....	23
Figure 7 - interoperability issues .....	26
Figure 8 - Extending the common data environment (CDE) .....	27
Figure 9 - Construction deficiencies related to the data.....	28
Figure 10- CDE functionalities .....	29
Figure 11 - Digital twin applications.....	31
Figure 12 - The four stages of the industrial revolution.....	33
Figure 13 - Industry 4 principles .....	34
Figure 14 - Core technologies of Industry 4.0.....	35
Figure 15 - Digital twin ecosystem .....	36
Figure 16 - Some IoT applications .....	38
Figure 17 - Abstraction of the IoT components .....	38
Figure 18 - IoT architectures, a) layered architecture—three and five-layer; b) middleware based; c) service-oriented based; d) fog based .....	39
Figure 19 - Three-layer IoT architecture .....	40
Figure 20 - Categorization of IoT devices.....	45
Figure 21 - Types of topology: gateway (GW), router (RT), and end device (FD) .....	46
Figure 22 - Cloud Stack .....	50
Figure 23 - IoT-Fog-Cloud connectivity .....	52
Figure 24 - A classification of the scope of fog computing and its related computing paradigms. (Intersection of cloud computing and mist computing is Cloud of Things, and the intersection of mobile computing and cloud computing is mobile cloud computing.) .....	53
Figure 25 - Comparison of fog computing and its related computing paradigms in terms of their location and distance from the core clouds. ....	55
Figure 26 - Cascina Nosedo project .....	57
Figure 27 - Window with movable louvers.....	58
Figure 28 - The workflow of collecting the environmental data (light status) and updating the BIM model in a real-time manner .....	59
Figure 29 - The workflow of getting the commands from the BIM model and sending them to the motor .....	60
Figure 30 - The 3D model of the physical model.....	61
Figure 31 - The physical prototype of the project's windows .....	61
Figure 32 - The embedded motor, boards, and light sensor in the constructed physical model .....	62
Figure 33 - The BIM model of Cascina .....	62
Figure 34 - Raspberry Pi board .....	63
Figure 35 - Raspberry Pi ports .....	63

Figure 36 - Raspberry Pi GPIO port.....	64
Figure 37 - LDR sensor .....	65
Figure 38 - The sample diagram of LDR connection .....	66
Figure 39 – The developed plugin by the writer for updating the DT model.....	67
Figure 40 - The developed plugin panels .....	68
Figure 41- Receiving options .....	68
Figure 42 - Updating the room parameters' values based on real-time data.....	68
Figure 43 - changing the room transparency based on the light intensity. a) light is not enough b) light is enough.....	69
Figure 44 - A window with light status .....	69
Figure 45 - Louvers' moving directions.....	70
Figure 46 - Parametric model of the window .....	70
Figure 47 - 28BYJ-48 Stepper Motor and ULN2003 board.....	71
Figure 48 - ULN2003 board.....	71
Figure 49 - Motor rotation mechanism.....	72
Figure 50 - The sample connection diagram of 28BYJ-48 and ULN2003 board to the Raspberry Pi..	72
Figure 51 - Scenarios for moving the louvers .....	73
Figure 52 - Summer Mode .....	74
Figure 53 - Winter mode .....	74
Figure 54 - Plug-in for updating the model and sending data to the physical model .....	75
Figure 55 - Enabling the VNC viewer in Raspberry Pi and adjusting the resolution.....	90
Figure 56 - Getting Raspberry Pi IP address .....	91
Figure 57 - Running the VNC viewer on a computer.....	91
Figure 58 - Script for reading the LDR data and writing in Google Sheets .....	92
Figure 59 - Script for reading the data from Google Sheets and store in local CSV file .....	93
Figure 60 - The script of "Receiving' panel written in pyRevit.....	95
Figure 61 - Script for modifying the model based on the selected scenario, getting rotation data from Revit, and writing in the local CSV file .....	97
Figure 62 - Python script for transferring data from a local CSV file to the Google Sheets.....	98
Figure 63 - Reading the commands from Google Sheets and sending to the motor via Python script in Raspberry Pi .....	99

## LIST OF TABLES

Table 1 - LOD system according to the different national specifications.....	25
Table 2 - industry 4 main pillars .....	34
Table 3 - Active and passive communication mode.....	41
Table 4 - Comparison between SaaS, PaaS, and IaaS.....	51
Table 5 - A list of computing paradigm acronyms used in this section .....	53

This page is intentionally left blank

## 1. INTRODUCTION

Building Information Modelling (BIM) is a digital representation of physical and functional characteristics of a facility that serves as a shared knowledge resource for information about a facility, forming a reliable basis for decisions during its lifecycle from inception onward (NBIMS, 2010). Because of BIM capabilities, it has also drawn the researchers' attention in the field of sustainability and green building (Dall'O', Zichi and Torri, 2020). Moreover, BIM can provide a valuable data set for Computer-Aided Facility Management (CAFM) systems (Karim *et al.*, 2018) and Building Management System (BMS). All these features are in the shadow of integration and full collaboration between project parties and different project phases.

Since designing and constructing in a digital environment is not too costly, BIM provides an opportunity for different analyses and optimization before construction. These analyses can affect directly the project objectives such as time, cost, quality, and HSE through the life cycle of the project.

It is noteworthy that the BIM model should not always be prepared with all the details, it should be based on the requirements since it is costly and make the model heavy. So, defining the required LOD in BIM Execution Plan (BEP) is crucial. Another aspect that should be considered is related to interoperability. Interoperability can be defined as the ability of one system (e.g. software application, workflow, process, ontology, etc.) to work with other systems without any effort on the part of the end-user of the system (Costin, 2016).

BIM has an imperative role in connecting the project stakeholders and integrating the process. The connection and collaboration of stakeholders such as owners, designers, general contractors, subcontractors, suppliers, and so on can be done in a Common Data Environment (CDE).

Generally, BIM is more focused on preventing errors and optimization in the design, construction, and maintenance and does not work with real-time data which is useful for monitoring and prediction in the facility management and operation phase. On the other hand, Digital Twin (DT) is a useful method that is capable of working with sensors and integrating real-time data which is categorized among industry 4 technologies and it is a good tool for prediction and automation.

DT is categorized as Cyber Security and Cyber-Systems in the industry 4 revolution that has brought many advantages in different industries. With the emergence of industry 4, different industries try to introduce smart products and high-end technological tools. The idea of building a twin refers to producing a copy of a part or product and using it for reasoning about other instances of the same part or product – thus establishing a relation between multiple copies. This idea is said to originate from NASA's Apollo program, "where at least two identical space vehicles were built to allow mirroring the conditions of the space vehicle during the mission"(Schleich *et al.*, 2017). The concept has been developed by the passage of time and now it is being used in various fields such as aerospace engineering, electric grid, car manufacturing, petroleum industry, healthcare, etc. (Tao *et al.*, 2019). The AEC industry and especially the facility management parties have focused on this technology as it can provide the opportunity to derive real-time data. This data which can be the result of a connection

between the virtual model and physical model through the sensors can be used for monitoring, improving energy efficiency, personalization, prediction, maintenance, and so on.

Sensors or generally the term IoT (Internet of Things) advancements in recent years have provided the opportunity to address real-time data availability. The main goal of this area is to connect the digital and physical entities that give a powerful tool to the decision-makers for better control and optimization. They are also widely used for monitoring and controlling the building's indoor environmental quality.

Generally, Indoor Environmental Quality (IEQ), describes the indoor buildings yardsticks that can improve occupant's satisfaction. IEQ considers air quality, access to daylight, views, acoustic conditions, occupant control over lighting, and thermal comfort. It also includes the building layout and access to different places. Several studies have investigated the effects of IEQ on productivity and its importance. (Geng *et al.*, 2017) (Sadick, Kpamma and Agyefi-Mensah, 2020)(Kallio *et al.*, 2020)(Rasheed, Khoshbakht and Baird, 2021)

As stated, IEQ is very important, so, monitoring, controlling, and automation can be beneficial for the building and entity. In contrast to the importance of daylighting that directly affects occupants' satisfaction and energy consumption, no study tries to design a process for using daylighting with the help of DT.

Implementing the DT requires a proper DT platform and different companies try to offer their solution as a DT platform. Although they have designed their platforms, most of them are expensive and require specialty for using them. So, this study aims to design a low-cost process and platform for monitoring, visualization, and automation of daylighting in a real-time status with the help of DT. The designed platform used the Autodesk Revit as an authoring software that can integrate the real-time data derived from Raspberry Pi and light sensor via the plug-in written in Python. This plugin can also send opening or closing commands to the windows for controlling the daylighting entrance from the Autodesk Revit or smartphone as an automation system. In the end, the capability of the offered model is tested in the case study.

This work is structured as follows. We briefly describe facility management in Section 2. Then we review the BIM concept and its application in Section 3. We highlight and discuss the digital twin concept and digital twin ecosystem in Section 4, also explaining the IoT and cloud computing in sections 5 and 6. We show the designed plugin and its application in our case study in section 7. Finally, Section 8 draws the conclusions.

## 2. FACILITY MANAGEMENT

### 2.1. Introduction

Based on IFMA, a leading association for facility management professionals, Facility Management (FM) “is a profession that encompasses multiple disciplines to ensure functionality, comfort, safety, and efficiency of the built environment by integrating people, place, process, and technology” (ifma, 2020). Also, International Standards Organization (ISO) defines FM as “an organizational function which integrates people, place, and process within the built environment to improve the quality of life of people and the productivity of the core business.” (International Organization for Standardization (ISO) 41011, 2017). With these definitions, it is clear that FM encompassed a large number of issues to benefit the organization and entity by considering the organization's objectives such as time, cost, quality, and HSE. Regarding functionality, FM should ensure the performance and reliability of the building or facility (Hyunji *et al.*, 2018).

Facility management has gained researchers' and industry parties' attention due to its importance. It is mentioned that 60-85 percent of the building's total life cycle cost relates to the operation and maintenance (Hilal, Maqsood and Abdekhodae, 2019). On the other hand, the population is on the rise in the world which needs more buildings and infrastructure. So, studying facility management can benefit a lot the Architecture, Engineering, and Construction, Facility Management (AEC/FM) industry.

In 2014, Cao *et al.* tried to improve the productivity, efficiency, and profitability of the facility management based on the occupants, satisfaction level. They used Artificial Intelligence (AI) in their framework that leads to a schedule based on available resources (Yang, Xinyi and Xuan, 2021). Having a standardized process of FM leads to an increment of efficiency. In 2018, Shin *et al.* developed a facility management process model of an office building. They introduced 6 main processes based on practical terms (Hyunji *et al.*, 2018). For improving facility management, researchers try to use the potentials of new technologies. For instance, Cong *et al.* studied the application of Radio Frequency Identification (RFID) to allow the easy and standardized exchange of building maintenance data among project partners (Z., L. and K., 2021)

In facility management, data availability about the assets plays an important role. However, many challenges exist regarding data insufficiency, comprising a lack of sufficient metering and accessibility and poor data quality (Gao and Pishdad-Bozorgi, 2020). To address this issue two terms “smart city” and “smart building” have been envisioned which be described in the next section. The smart building which is a fundamental part of the smart city requires intelligent and integrated data that can be used during asset utilization.

To achieve this, different systems emerged such as Building Automation System (BAS) or Building Management System (BMS), Computerized Maintenance Management System (CMMS).

Based on functionality and comfort, different criteria can be considered. These criteria which are known as Indoor Environmental Quality (IEQ) can be energy use, temperature, Humidity, and so forth.

So, the following sections will be discussed these terms and will reflect the latest research results.

## 2.2. Indoor environmental quality

“The quality of buildings can be assessed in terms of the indoor air quality, thermal comfort, lighting quality, acoustic comfort afforded the occupants, collectively referred to as Indoor Environmental Quality (IEQ)” (Parkinson, Parkinson and de Dear, 2019). Due to the growth of population and highlighted the effect of IEQ on life quality, work productivity, energy consumption, and cost, this topic has gained attention during the last decades. IEQ can be studied in different aspects. first of all, understanding the importance of these factors and identifying them, and second the technologies and methods to monitor and control the IEQ parameters in the built environment.

Kim et al. tried to find the important factors which affect the occupants' satisfaction and defined their relative importance. They categorized the 15 selected criteria through the questionnaire and understood that amount of space, noise level, visual privacy, amount of light, air quality, and sound privacy are highly affects occupants' satisfaction (Kim and de Dear, 2012). In another study, Korsavi et al. focused on the four groups of thermal comfort, indoor air quality, visual and acoustic comfort and they worked on the IEQ parameters impact on children's overall comfort. They found that air sensation and operative temperature are the most effective criteria in children's satisfaction (Korsavi, Montazami and Mumovic, 2020).

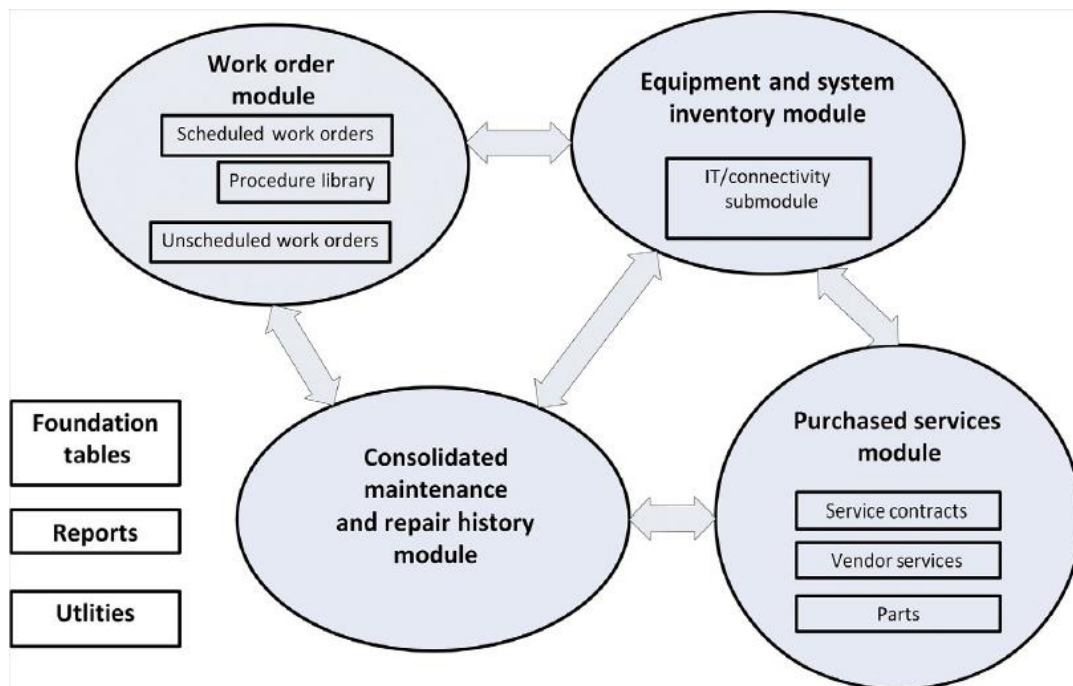
IEQ parameters have a direct impact on work productivity. Sadick et al. showed that the improving IEQ factors of offices have a positive impact on job satisfaction and productivity. In their study, acoustic and privacy satisfaction, ventilation and thermal comfort, and lighting satisfaction were considered as IEQ parameters (Sadick, Kpamma and Agyefi-Mensah, 2020). In a similar study, Ji et al. investigated and measured IEQ factors such as air temperature, globe temperature, relative humidity, carbon dioxide concentration, illuminance, and background noise. They also found that the relative productivity could be predicted based on air temperature, thermal satisfaction,

The above research and a lot of other researches show the importance of IEQ factors. So, different technologies and methodologies try to measure, monitor and predict these factors and their effects. Moreover, as a final solution for a thorough and effective interaction with the built environment, automating the process of adjusting these factors in the building environment based on desired objects is triggered.

## 2.3. Computerized maintenance management system

The huge amount of information in the operation and maintenance phase cannot be managed without computers. A Computerized Maintenance Management System (CMMS) is a computer-based solution for gathering and handling information for reducing the cost of improving the quality of service. There are a lot of indirect costs in equipment maintenance (up to 5 times of direct costs) such as lower quality, higher energy, reduces asset life, wasted resources, late deliveries, environmental issues, and safety risks (Wienker, Henderson and Volkerts, 2016). To optimize these costs and improvement in quality of service, planning preventive maintenance actions, scheduling of activities considering available resources and planned production, management of spare parts, analysis of data to reduce the occurrence of failures, and improving the performance of the maintenance function are the measurements should be considered (Lopes *et al.*, 2016). It is stated that after CMMS implementation, the productivity of the

maintenance sector can increase from 35% to 70%-80%, and inventory costs can reduce by 5% to 12% (Cato and Mobley, 2002). In summary, each CMMS has 4 main parts which are depicted in Figure 1 by Cohen et al. (Cohen, Baretich and Gentles, 2020):



**Figure 1 - Simplified CMMS module structure**

- Equipment inventory: is an automated file of all the equipment that has been included in the CMMS.
- Consolidated repair and maintenance history: is a record of each repair and maintenance event, independent of who initiated the event and who provided the service.
- Work order control: is used to dispatch and prioritize requested work, schedule periodic inspections, and preventive maintenance, and track the status of pending scheduled and unscheduled work orders
- Purchased services module: This section describes various ways to track and manage purchased repair parts and vendor services.

Different researchers try to focus on one or more than one field mentioned above and all try to find the solution for managing the information and reducing the data transferring time to the desired destination. Munyensanga et al. focused on scheduling the preventive maintenance tasks. They presented the android-based platform to address the delay communication in the workstation and maintenance field (Munyensanga *et al.*, 2018). Their platform can mutate the handwriting to the interface and storing the data in the database. Dashboard, work order, checklist, completed, assignment, and library are parts of their application interface.

There are a lot of CMMS applications in the market that should be selected and be used based on the requirements. Lopes et al. tried to present a decision-making platform for selecting the proper CMMS

software based on Analytical Hierarchy Process (AHP). They considered 5 main criteria: performance, usability, compatibility and portability, security, and supplier-related factors, and also 16 sub-criteria.

In summary, the computer as the main tool for managing the information is being used by researchers for improving the operation and maintenance phase. They also combine it with other technologies and concepts such as BIM, Augmented Reality (AR), Virtual Reality (VR), digital twin, and so forth. However, all of them try to improve the core activities in CMMS.

#### **2.4. Building Automation System (BAS) or Building Management System (BMS)**

Building Automation System (BAS) or Building Management System (BMS) is a computer-based system for monitoring and controlling the building's electrical and mechanical equipment such as power system, lighting, and ventilation to confirm sustainability (Hossain, 2019). The BMS is important for both energy consumption and sustainability as well as occupants' satisfaction. It is mentioned that buildings account for around 40% of the world's total energy use (Yang *et al.*, 2020). So, any saving in this part leads to saving a huge portion of total energy consumption. This reduction in energy consumption also helps the green building concept and Nearly Zero Energy Building (NZEB) and sustainability due to a decrease in the amount of CO<sub>2</sub> emission and low energy consumption. It is noteworthy that the managerial issue and organizational support for adapting and using these technologies is important as well as technological issues (Maqsoom *et al.*, 2020)

BMS tries to use different technologies and devices such as sensors to access the proper data and manage this information to make a balance and reach the goals. For this achievement, a structured network of equipment and sensors are required which can be improved by BIM and digital twin. Some of these concepts, technologies, and procedures will be defined in the following chapters.

Different researchers try to study BMS and use it for improving the buildings' functionality in case of energy consumption and occupants' satisfaction. The BMS data can also be used for other analyses and energy consumption prediction. Ghani *et al.* used the data derived from the BMS system for energy prediction with Artificial Neural Network (ANN). Ghani *et al.* believe that saving energy in a time of low energy demand is a good and imperative task which have a dramatic effect on total energy consumption. They also make a device, a solar photovoltaic terrarium for saving energy (Ghani *et al.*, 2020). Because of dealing with a large amount of data, managing and handling the information is crucial. Zhan *et al.* tried to organize the metadata of BMS and mapping the data points for creating the Building Energy Modeling (BEM). They tried to facilitate the process of interpreting the derived information from BMS for further energy analysis. They also used BIM and Industrial Foundation Class (IFC) extension formats in their work (Zhan, Chong and Lasternas, 2021).

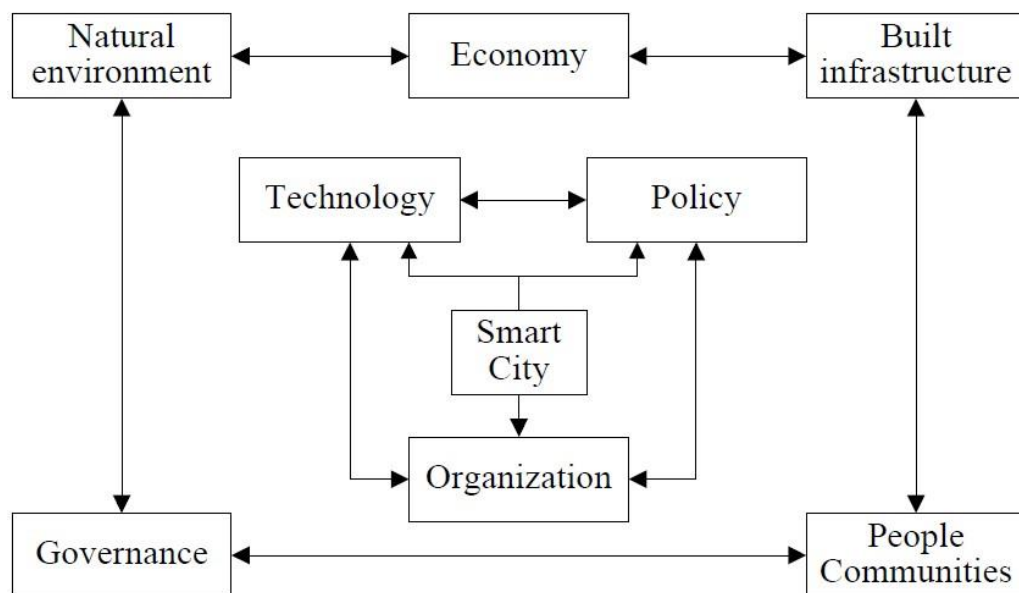
These systems are fundamental moves toward smart buildings and smart sustainable cities which will be discussed in the next sections.

#### **2.5. Smart building and smart city**

The growth of population and rapid urbanization requires extra consideration. It is predicted that the percentage of people living in the urban area will reach 68% by 2050 from 55% in 2018 which has

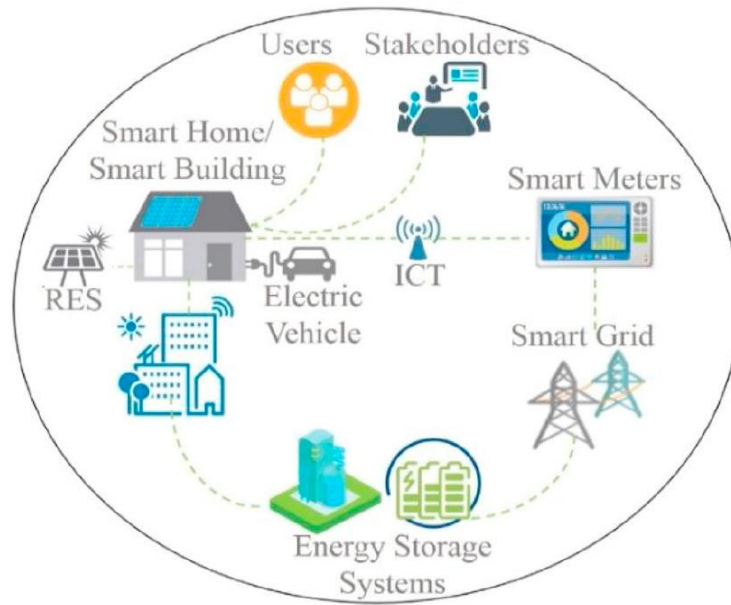
significant growth in comparison to 1950 (30% in 1950) (United nations, 2019). So, new strategies and management should be deployed to overcome the urbanization consequences such as required infrastructure, sustainability, health, resource demand, waste management, transportation, and so forth. Accordingly, the term and concept of Smart City (SC) was introduced and has gained popularity. “Intelligent”, “digital”, “smart”, “sustainable smart”, or “smart sustainable” are the terms used and changed over time in this field (Apanaviciene, Vanagas and Fokaidis, 2020). Although there is no consensus about the SC definition, “it is an agenda aiming to implement technological innovations and utilize digital data collected about society as a means of policymaking and urban development” (Michalec, Hayes and Longhurst, 2019). Not only energy efficiency is important in SC, but also it requires the adoption of remote monitoring and control systems (Roche *et al.*, 2012).

Implementing the smart city requires different factors that play role in creating the smart city. Based on the proposed model by Li, the fundamental factors that impact smart cities are shown in Figure 2 (Li and Liu, 2013):



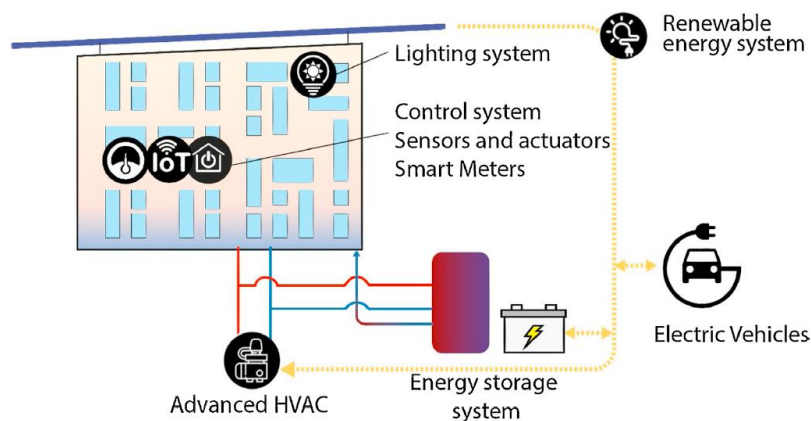
**Figure 2 - Factors in the framework of Smart City**

Indeed, technology, policy, and organization shape the smart city and the others, economy, built environment, people communities, governance, and natural environment are not only the components of a smart city but also are affected by a smart city. In summary, smart city components are illustrated by Dakheel *et al.* in Figure 3 that shows the connection between smart houses, smart grids, and other components which can be done via Information and Communications Technology (ICT) (Al Dakheel *et al.*, 2020).



**Figure 3 - Smart City Components**

In the smart city ecosystem, construction plays a crucial role in reaching the smart city. So, constructing the new building and infrastructures or refurbishments should follow smart city features. The term “smart city” emerged to meet the requirements. So, smart buildings are considered to be one of the fundamental components of the smart city ecosystem. Especially in the operation phase, smart building enables interaction between different services and also make it possible to share information with other buildings that provide good data for better decision making at the city management level. When the data is collected from different sources in the smart city management platform, the most efficient decision can be made in favor of the building as well as the dwellers of the city. This opportunity can be provided via different sensors and actuators that can monitor, control, and automate services such as heating, air conditioning, ventilation, lighting, water, security, and so forth. Generally, different technologies are shown by Dakheel in Figure 4 (Al Dakheel *et al.*, 2020). So, IoTs are new devices that can provide useful real-time data. We will discuss more about IoT in the following sections.



**Figure 4 - Key Technologies in Smart Buildings**

## 3. BUILDING INFORMATION MODELING (BIM)

### 3.1. Introductions

Building Information Management (BIM) is an umbrella term describing production and management processes in which construction procedures, as well as physical and functional characteristics of buildings, are represented digitally before they are committed to physical space and usage.

BIM is a term in the AEC industry that we can hear in every project. The acronym stands both for building information modeling (the process) and building information model (the artefact) and also it can stand for building information management and building information marketing. (Turk, 2016). BIM is typically employed to predict and control construction procedures and building performance to minimize cost in terms of energy, materials, and labour (Herr and Fischer, 2019). Generally, BIM comprises technologies and processes that can improve the efficiency of the whole life cycle processes of the project from design to demolition phases. Based on Penn State Standard, there are more than 20 uses for BIM for the phases of plan, design, construct, and operate. Figure 5 shows some of these uses but not limited to them.

Because of BIM capabilities, it has also drawn the attention of researchers in the field of sustainability and green building (Dall'O', Zichi and Torri, 2020). Moreover, It is stated that BIM can provide a valuable data set for CAFM systems (Karim *et al.*, 2018) and FM. All these features are in the shadow of integration and full collaboration between project parties and different project phases.

Since designing and constructing in a digital environment is not too costly, BIM provides an opportunity for different analyses and optimization before construction. These analyses can affect directly the project objectives such as time, cost, quality, and HSE through the life cycle of the project.

As a misperception, it is thought that BIM is equal to the 3D model or 3D model authoring software such as Revit. Embedding the information in the 3D model is a key difference that distinguishes a 3D model from a BIM model. Based on Eastman definition (Eastman *et al.*, 2018), BIM models are characterized by:

- Building components that are represented with digital representations (objects) that carry computable graphic and data attributes that identify them to software applications, as well as parametric rules that allow them to be manipulated intelligently.
- Components that include data that describe how they behave, as needed for analyses and work processes, such as quantity takeoff, specification, and energy analysis.
- Consistent and nonredundant data such that changes to component data are represented in all views of the component and the assemblies of which it is a part.

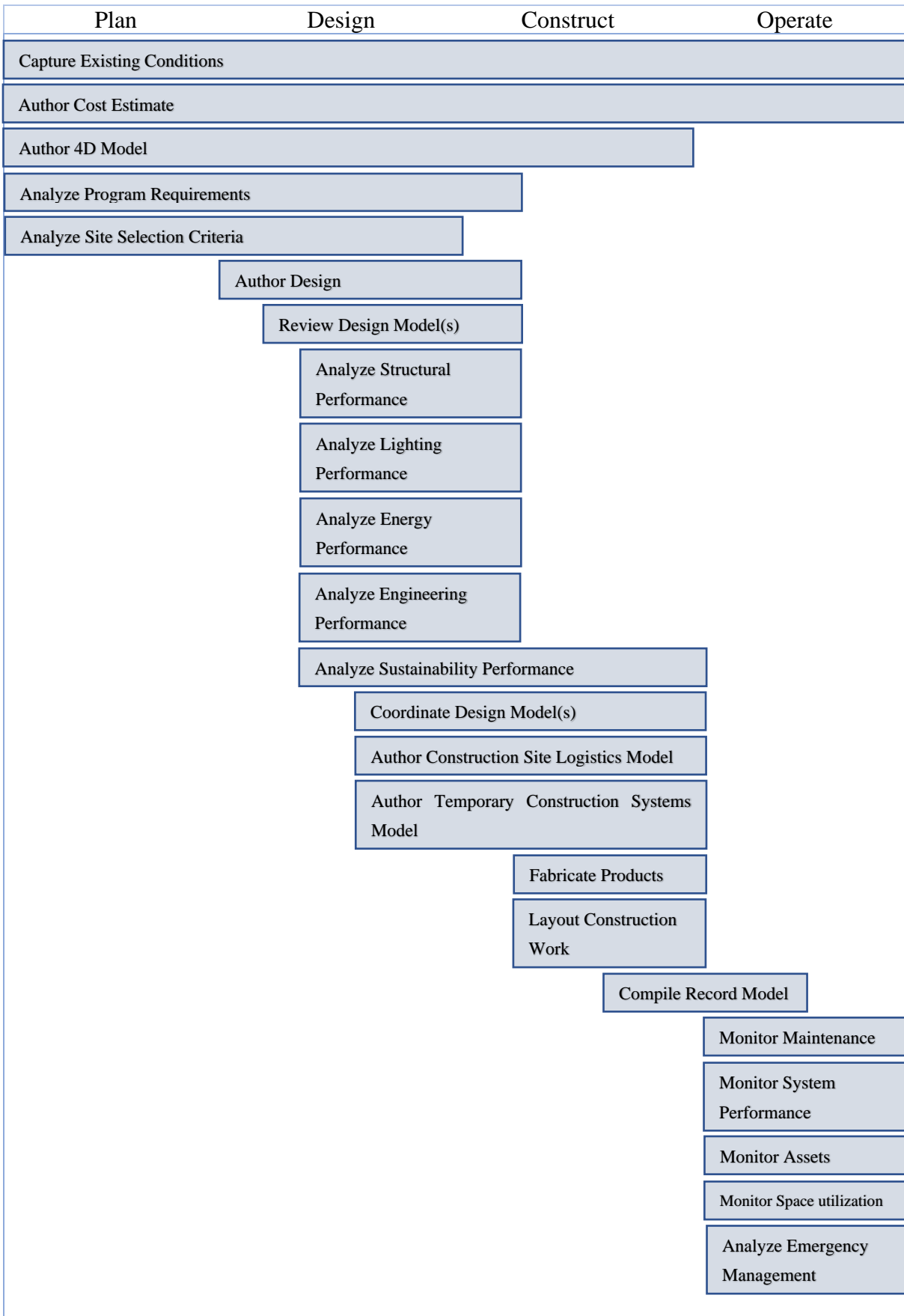
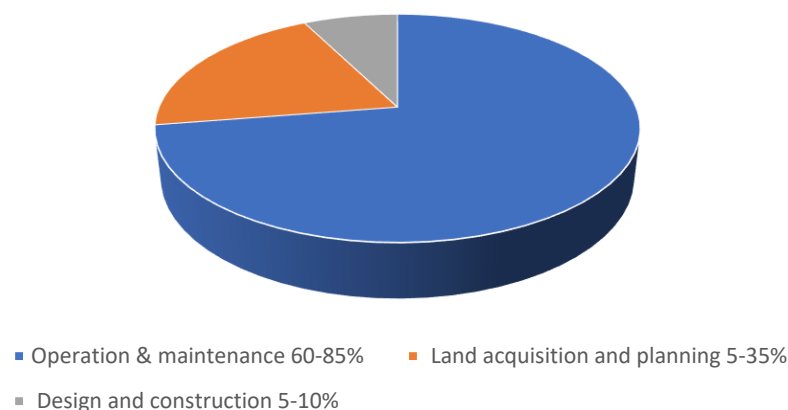


Figure 5 - BIM uses

### 3.2. BIM and Facility management

In the last decades, BIM widely used in the design and construction phase. Due to the importance and cost of facility management in the project life cycle, researchers and different parties try to benefit from BIM in facility management. According to the Terreno et al. research, BIM plays an imperative role in project success based on critical success factors (Saratu, Chimay and Somayeh, 2021). Based on the building typology and functionality, operation and maintenance constitute 60%-85% of the total project life cycle costs (Figure 6) which also can be changed based on the project type and location. (Aziz, Nawawi and Ariff, 2016). So, any improvement in the operation and maintenance phase with new processes and technologies such as BIM can improve the project's life cycle goals dramatically.

Life cycle cost budget division of a building



**Figure 6 - life cycle cost budget division of a building**

There are different topics in the field of facility management that have taken researchers attractions. Statistically, based on the issued articles, Liu et al. found the important topics in the field of BIM use in facility management. These categories in order of importance are as follow: (Hexu, Osama and William, 2021)

- As-built/as-is data acquisition
- Interoperability
- Sustainability
- Life-cycle collaboration and integration
- Information communication and visualization
- Knowledge-based facility management

Generally, BIM has the potential to address the challenges such as (Pardis, 2017):

- Information accessibility issues
- localization of the system or asset in large and complex buildings
- Determination of which facilities' spaces or systems will become affected as a result of a current malfunctioning or ongoing repair operation.

These challenges are crucial and dramatically affect FM efficiency. For instance, around 50% of facility managers' time is spent on locating the problems and identifying building elements that need inspection and control when using the information provided by the available FM systems (Koch et al., 2014). To

overcome these challenges, a structured FM system is needed that comprises facility planning, maintenance, procurement, cleaning, support services, property services, catering, and security services (E. et al., 2021).

Continuous availability of reliable information about asset inventory, condition, and performance are integral parts in FM (Alvarez-Romero, 2014) which can be handled by BIM through providing an effective data source for FM (Volk, Stengel and Schultmann, 2014).

There are different efforts in the FM field conducted for improving efficiency. In 2020, Ergan et al. tried to mirror some indicators such as thermal comfort, indoor air quality, acoustic comfort, visual comfort, building design, and building services in the BIM model. In their proposed model, these data are derived from occupants' feedbacks that are imported in the model based on locations. They selected the IFC extension as a repository for federating and visualizing the information. To implement the semantic data model in the IFC schema, they used *IfcPerformanceHistory* and *IfcAnnotation* entities for capturing, localization, and visualization of information. So, they collect and formalized the unstructured data of feedbacks which can be useful for decision-makers. Occupant feedback regarding building conditions is an important type of FM information for decision-makers, sometimes it is hard to make consensus in public places as they are used by different people with different features. (E. et al., 2021)

In another research, Wang et al. tried to investigate the facility risks and strategy actions at the level of a single piece of a component. In their BIM-AR platform, the Analytic Hierarchy Process (AHP) approach was used to determine whether a single piece of equipment should have corrective, periodic, or condition-based maintenance. Also, Multiple-Criteria Decision-Making (MCDM) is used for prioritizing the equipment components which can help decision-makers for facility management. (Ting-Kwei and Yanmei, 2019)

Since asset management and facility management require information that should be prepared in different phases, Farghaly et al. tried to provide a platform for necessary data. They synthesized the nongeometric data and developed the taxonomy that comprised 6 main groups: classification, space/location, asset CAPEX, warranty, maintenance, and specifications. They also created a plug-in for Revit which can be used for adding the required parameters to each Revit object that can be defined as an object. (Karim et al., 2018)

Having the updated data is a key factor in FM that can be useful for decision making. This issue leads to try for having real-time data and the ability for real-time interaction with facilities and equipment. This attitude is considered for designing a DT model and real-time interaction in this study that will be described later.

### **3.3. LOD**

As we described, embedding information in a 3D model is a fundamental part of the BIM process. Meanwhile, the granularity and amount of graphical and non-graphical information are challenging. On one side, having a precise model with a lot of graphical and non-graphical information is useful and can help the project in different phases. On the other hand, not only providing such an enriched model is time-consuming and costly, but also they are very heavy and cause trouble for using them. So, first, we

have to have a clear understanding of the level of graphical and non-graphical information and then decide which one is useful in our project based on the model uses that should be reflected in BEP. The clear definition of LOD levels and expected output in each phase of the project leads to reduce the risk of miscommunication among the project's members. So each party knows that with what quality they have to deliver the model and what they expected to receive from other parties.

Different countries have tried to provide standards about LOD. The abbreviation LOD was first introduced by the USA (BIMFORUM) as a Level of detail that was related to the graphical features. Then, this concept was completed and LOD was introduced as a Level of development. Today, LOD is used in different meaning by different countries, USA (BIMForum Specification), UK (BS 1192-1 and PAS 1192-2, 3), and Italy (UNI 11337 part 4) (Karlapudi et al., 2020), which is briefly shown in Table 1.

**Table 1 - LOD system according to the different national specifications.**

Country	LOD means	Sub-type	Scale
USA	Level of Development	LOD: As Designed	LOD 100, LOD 200, LOD 300, LOD 350, LOD 400
		LOD: As Built	LOD 500
UK	Level of Definition	LOD: Level of Detail	LOD 1, LOD 2, LOD 3, LOD 4, LOD 5, LOD 6
		LOI: Level of Information	LOI 1, LOI 2, LOI 3, LOI 4, LOI 5, LOI 6
Italy	Level of Development of Objects	LOG – Geometrical Objects	LOG A, LOG B, LOG C, LOG D, LOG E, LOG F, LOG G
		LOI – Information Objects	LOI A, LOI B, LOI C, LOI D, LOI E, LOI F, LOI G

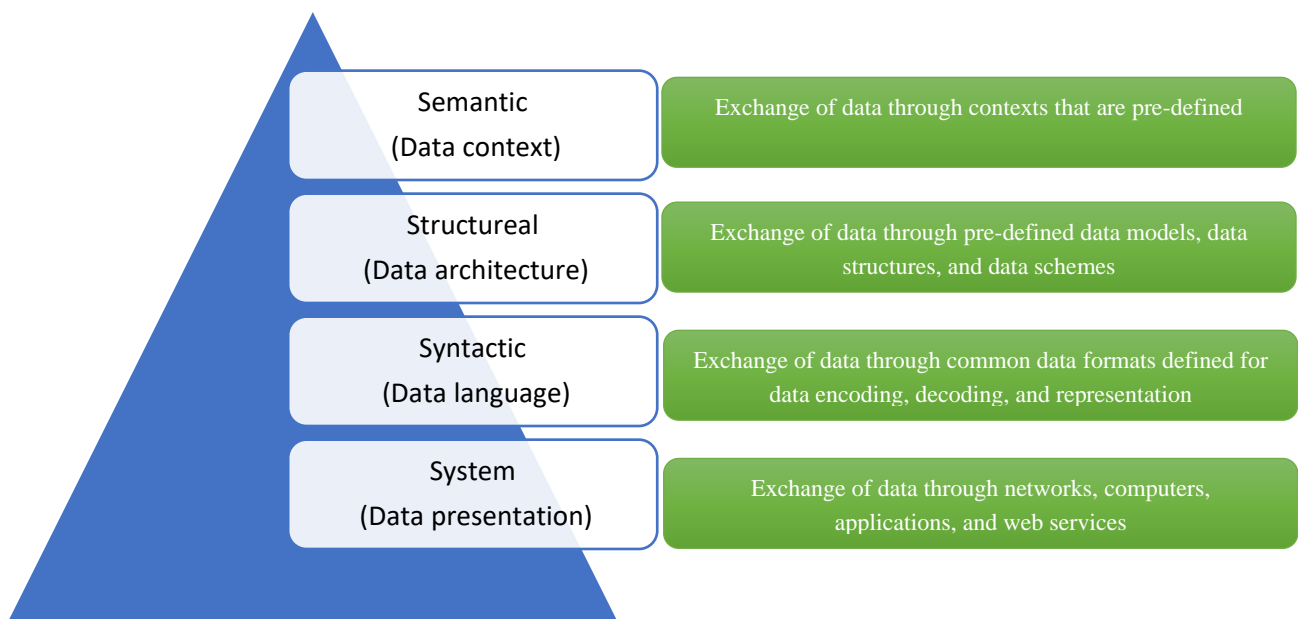
By the publication of ISO 19650 part 2, the new concept emerged as Level Of Information Need (LOIN). “The LOIN aims to replace the model of predefined scales and to switch into a gradual, incremental, milestone-free system with relevant points established beforehand. However, the LOIN concept is under development (Karlapudi et al., 2020).

### 3.4. Interoperability

Interoperability is a challenging topic in the BIM field. It is always on the list of barriers to BIM adoption in the AEC industry (Reza et al., 2015) (Ullah, Lill and Witt, 2019)(Saka and Chan, 2020). So, understanding this concept and providing the solution for interoperability improvement can help BIM implementation.

The word interoperability comprises two words of “intro” which means among and between and “operability” which means the ability to keep a piece of equipment, a system, or a whole industrial installation in a safe and reliable functioning condition, according to pre-defined operational requirements (Wikipedia, 2021). Based on ISO 25964 -2:2013, interoperability is the “ability of two or more systems or components to exchange information and to use the information that has been exchanged” (Zeng, 2019). In another definition, it is defined as the ability of multiple systems with different hardware and software platforms, data structures, and interfaces to exchange data with minimal loss of content and functionality (Guenther and Radebaugh, 2004). Although the concept of interoperability is a wide definition and can be applied to business, organization, projects, and people, the focus of BIM interoperability is mostly on information and technical interoperability.

As it is depicted in Figure 7, interoperability has four layers (Zeng, 2019):



**Figure 7 - interoperability issues**

As one of the efforts to solve interoperability in the BIM environment, buildingSMART introduced IFC for the exchange of BIM models between different software. However, interoperability is still low and needs progress.

### 3.5. CDE

Due to the collaborative nature of BIM, defining the process and technologies that can facilitate the connection between project stakeholders has received increasing attention. Since a lot of information

with different revisions is being produced, the trustable environment should be deployed to ensure the data management issues such as structuring, combining, distribution, and archiving. Common Data Environment (CDE) is a solution offered by standards such as PAS 1192-2:2013 that described this process for ensuring the data exchange quality. CDE is a “single source of information for any given project, used to collect, manage and disseminate all relevant approved project documents for multi-disciplinary teams in a managed process” (BSI Standards Limited, 2013)

Based on PAS 1192-2:2013, we have 4 main stages which are shown in Figure 8:

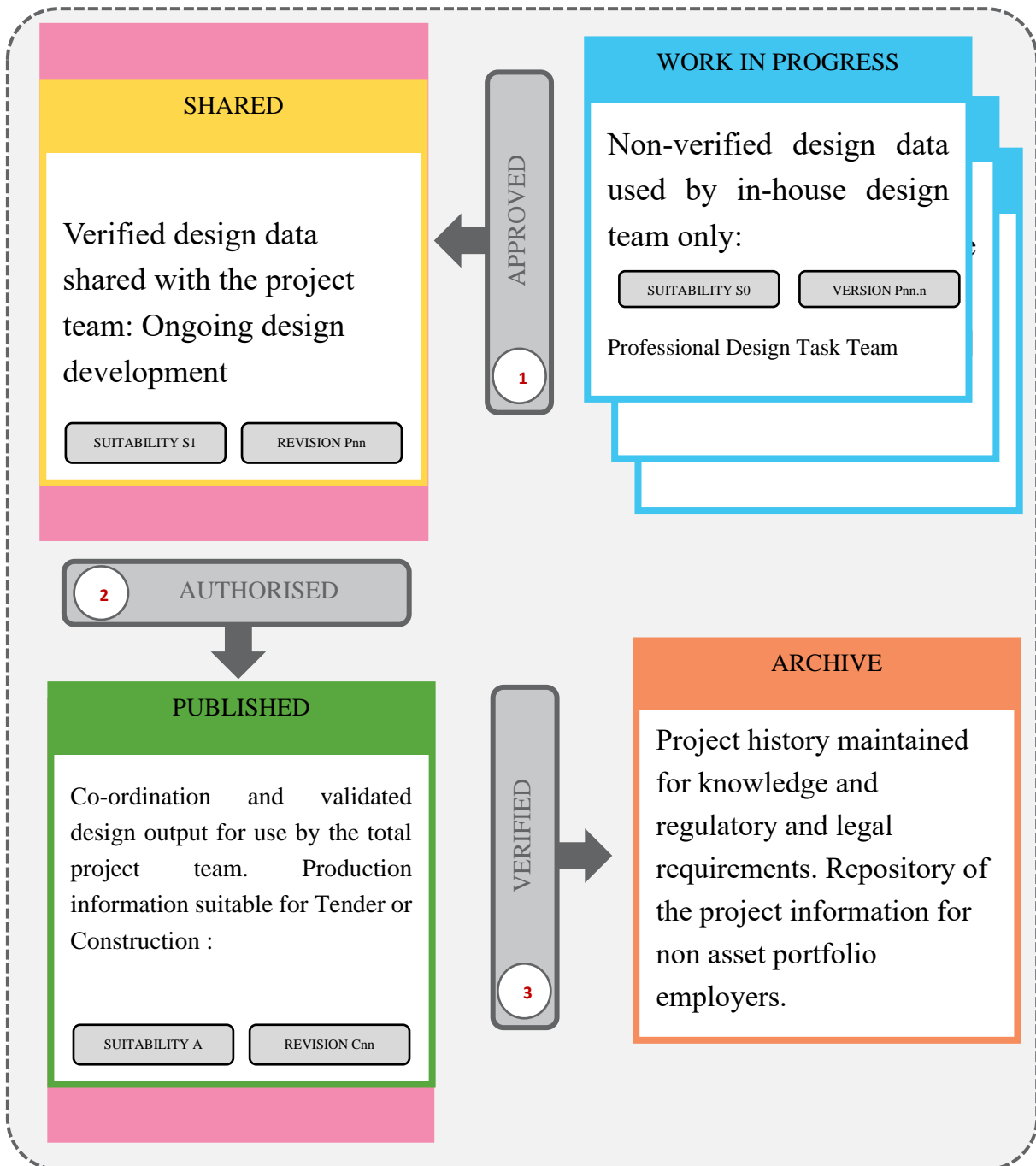
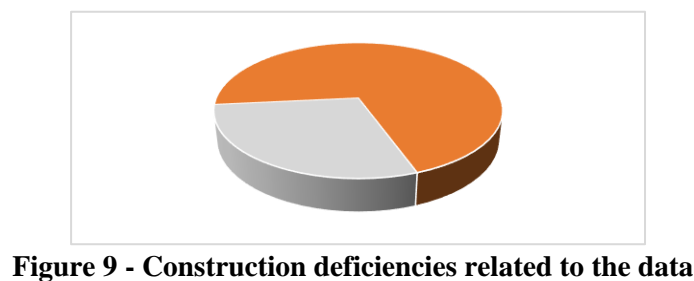
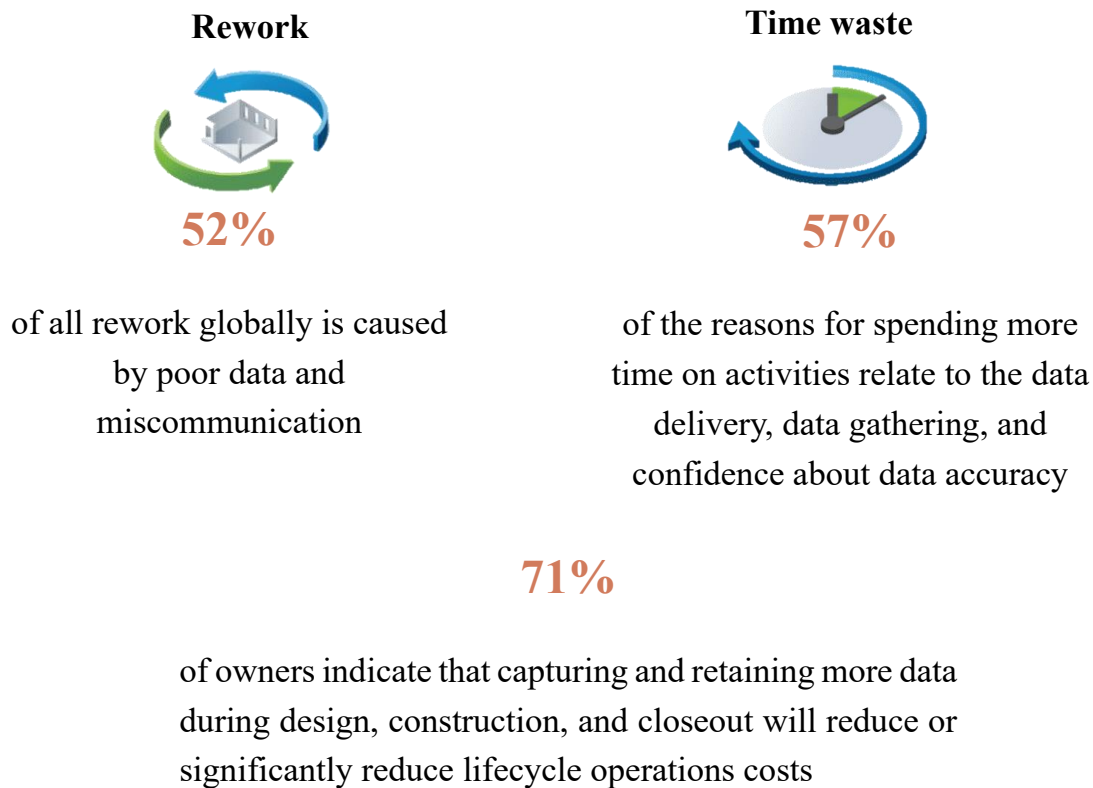


Figure 8 - Extending the common data environment (CDE)

Using the CDE in projects leads to efficiency improvement. Based on PlanGrid survey, we understand that data availability and communication play a fundamental role. Some of the deficiencies rates related to data are shown in Figure 9 (PlanGrid, 2018):



Different companies such as Acca, Autodesk, Trimble, and so forth have offered their platforms with different modules. Although they have some differences, they mostly have similar functions. Some of these functions are shown in Figure 10.

To sum up, selecting a proper CDE based on the project’s requirement and using that in the project is one of the first measures that should be taken in the project to improve accuracy, efficiency, project economy, accountability, and attractiveness.

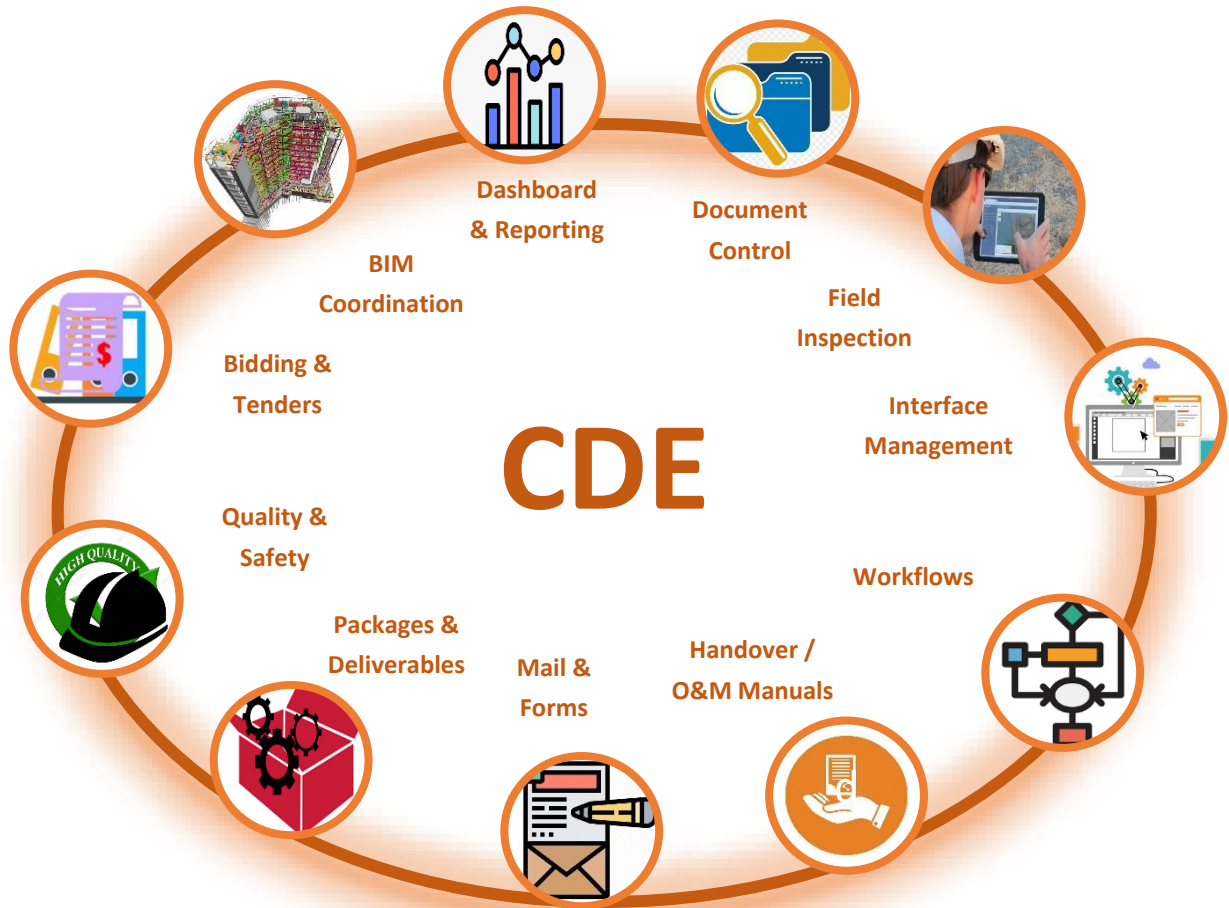


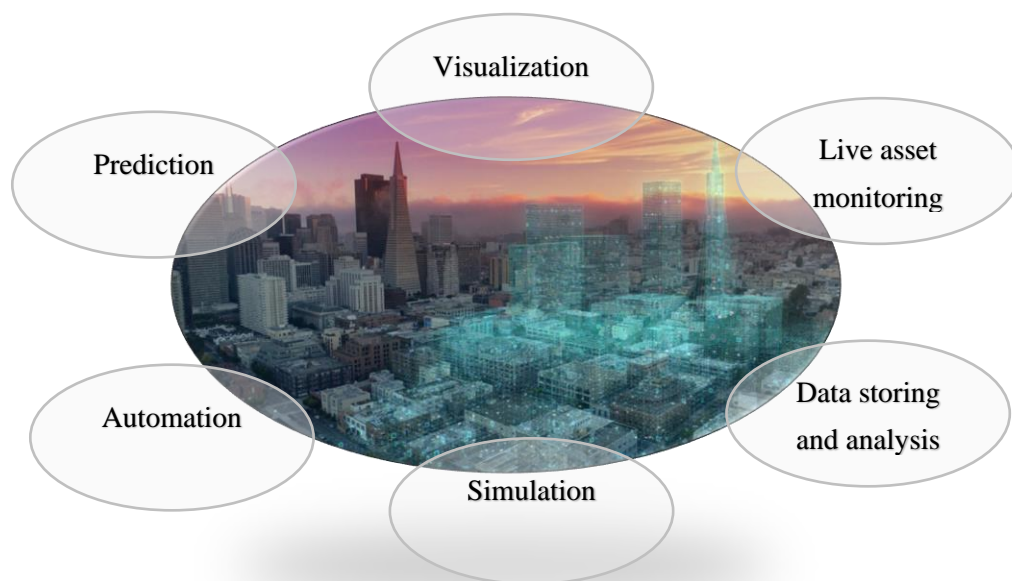
Figure 10- CDE functionalities

This page is intentionally left blank

## 4. DIGITAL TWIN

### 4.1. Introduction

Digital Twin (DT) is part of the industry 4 revolution which brought researchers and industry party's attention. It is a new concept that is under development. Every day, a new application will be industrialized based on its features. There are a lot of definitions offered by researchers (Semeraro *et al.*, 2021), however, all of them have common issues and parts: the physical product, the virtual product, and the connection between them. Generally, DT is invented to provide the situation to reflect the status of the physical object in a real-time situation on the virtual model. This combination which is called the DT model can be used for different purposes (Figure 11).



**Figure 11 - Digital twin applications**

Although DT first was used in NASA's Apollo program, "where at least two identical space vehicles were built to allow mirroring the conditions of the space vehicle during the mission" (Schleich *et al.*, 2017), today it is using in different fields such as aerospace engineering, electric grid, car manufacturing, petroleum industry, healthcare, etc.(Tao *et al.*, 2019). Ansys, Dassault, Bosch, GE, IBM, Microsoft, PTC, SIEMENS, and TESLA are leading companies that utilizing the DT for improving their business and solve challenges (Aheleroff *et al.*, 2021)

The border between the BIM model and the DT model is sometimes misunderstood. The BIM model that is described in section 3, mostly tries to optimize and facilitate the process of design and construction without considering the actual data in the construction site or information from equipment and asset in the operation phase. Although it is capable to provide a lot of useful data even for facility management in the operation phase, it can not handle and work with real-time data. Indeed, the BIM model is considered as an input and it is one of the important parts of the DT model. By connecting the

BIM model to the real world and physical assets, the DT model is constructed which can provide a valuable tool for facility management.

Like BIM models, the granularity of information, level of details, and derived data from the real world depend on the model purpose. It is worth mentioning that since modeling and especially IoT installation is costly, the model development should be aligned with the model's purposes. In addition, in the IoT ecosystem, a lot of data is being produced that requires management. So, redundant data make this process difficult.

There are a lot of studies that used DT in the Architecture, Engineering, and Construction (AEC) industry. For instance, Lu et al. tried to implement DT in the field of Operation and Maintenance (O&M). They used the DT model for managing the building pumps by installing the vibration detector sensors. Analyzing the outputs, let them decide about the pump status. They also used the Bayesian Online Changepoint Detection (BOCPD) method for analyzing the result from the sensors (Lu *et al.*, 2020). Even in recent researches, researchers try to use the DT in the construction process. Pan et al. provide a system for providing the construction progress information via Unmanned Aerial Vehicle (UAV). They integrated these real-time data with the 3D model (the IFC file) and tried to use this DT for construction management and progress prediction (Pan and Zhang, 2021). Also, the DT can provide a base for real-time project information sharing during the construction phase which is done by Lee et al. in 2021 (Lee *et al.*, 2021). They also used the blockchain framework for data storage, security, and sharing.

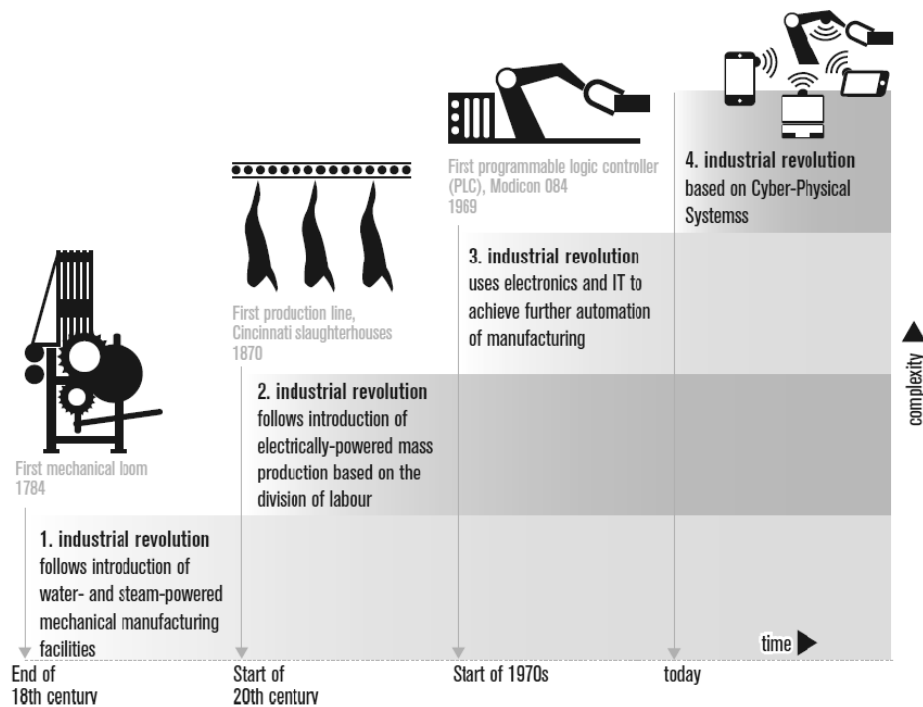
In contrast with the importance of lighting, few studies employ DT for improving lighting. In 2019, Duplákóvá et al. tried to verify their simulation software (Evo 6.1) by comparing the real data derived from the DT model and the results of the simulation software. Then they used it for further analysis and optimized the lighting intensity and analysis (Duplákóvá *et al.*, 2019). Based on the Onile research which done as a review in the field of energy consumption, most of the researchers in the field of lighting pay attention to the technological aspect and there is no study for managing the lighting system by DT with considering the daylighting potentials (Onile *et al.*, 2021). So, the current research will focus on creating an automated DT-based platform for optimizing daylighting by considering occupants' comfort.

The following parts in this section will discuss Industry 4 revolution and DT ecosystem and platforms

## **4.2. Industry 4**

Industrial revolutions show the changes in human lifestyle that transformed from agrarian and rural societies to the present. These changes reflect the changes in technologies and people's lifestyles. Studying history empowers us to a better understanding of the process and it can also give us a clue about future technologies and trends.

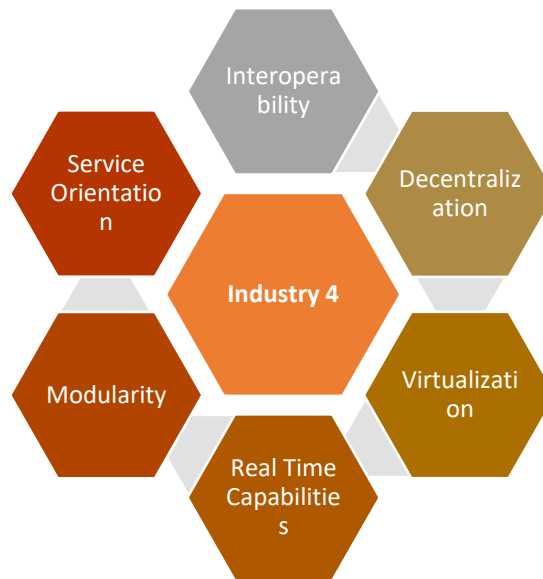
Generally, 4 stages are described for industry revolutions and now we are on the 4<sup>th</sup> one. As it is depicted in Figure 12, these stages are as follow (Alaloul *et al.*, 2018)(Vaidya, Ambad and Bhosle, 2018) (Tinmaz, 2020):



**Figure 12 - The four stages of the industrial revolution**

- Industry 1: The epoch that steam power was harnessed to run the factories and moving toward mechanization
- Industry 2: the era of electricity introduction to the industry that led to the development of production lines and efficiency and mass production
- Industry 3: The stage of implementing computer technologies leading to automation
- Industry 4: The current revolution stage that relates to the Cyber-Physical Systems (CPS) and connectivity in everyday life

Industry 4 which has brought significant contributions, has the following principles (Figure 13)(Carvalho *et al.*, 2018).



**Figure 13 - Industry 4 principles**

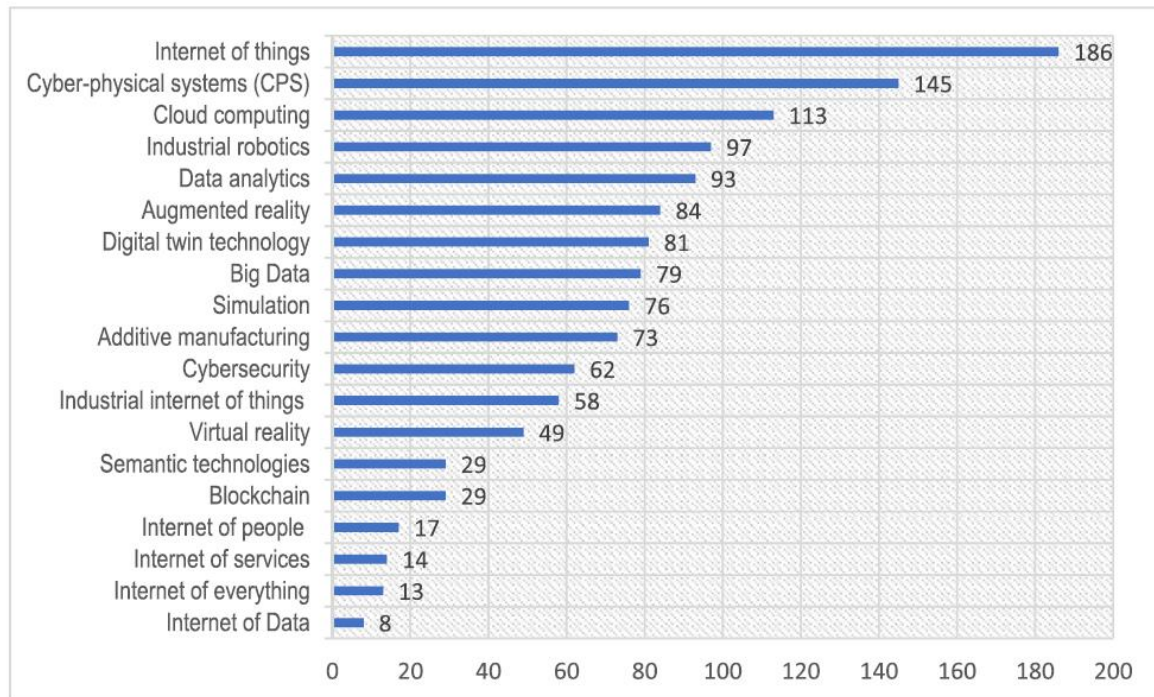
Also, industry 4 has 9 main pillars (Table 2) (Vaidya, Ambad and Bhosle, 2018).

**Table 2 - industry 4 main pillars**

Big Data and Analytics
Autonomous Robots
Simulation
System Integration
The Industrial Internet of Things
Cyber security and Cyber Physical Systems (CPS)
The Cloud
Additive Manufacturing
Augmented Reality

There are a lot of topics that have attracted the researchers' attention around industry 4 issues. Based on the Ghobakhloo et al. study, the core technologies of industry 4 which are in development are shown in Figure 14 (Ghobakhloo *et al.*, 2021). In this chart, the values reflect the numbers of the article that have allegedly acknowledged these technologies as building blocks of Industry 4.0.

Base on their research, they found that the internet of things, cyber-physical systems, and digital twin technology are higher-tier technologies.



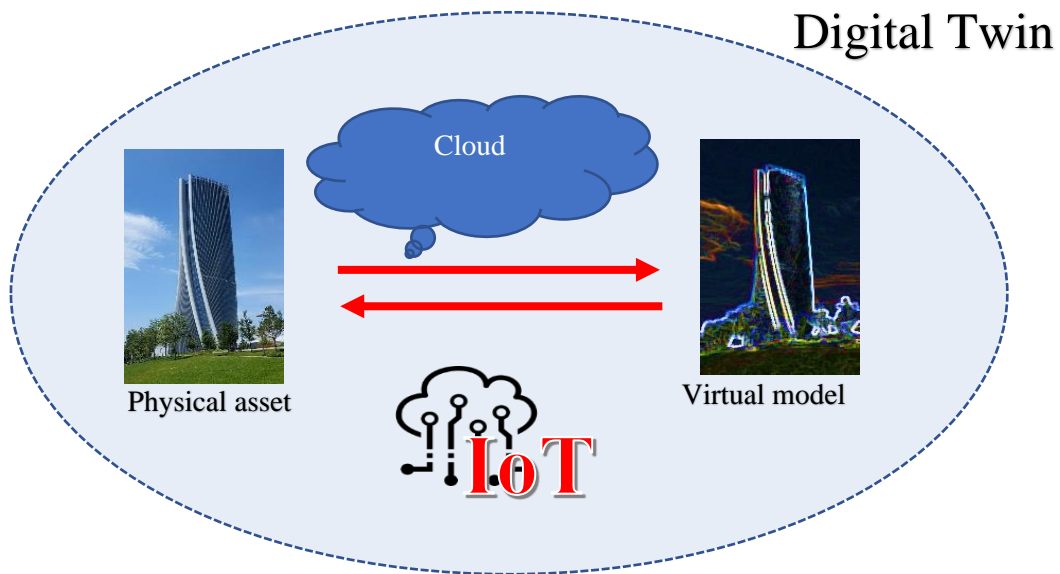
**Figure 14 - Core technologies of Industry 4.0.**

In comparison to other industries, the AEC industry is behind in the use of these new technologies. Indeed, AEC has great potential to update method statements and working workflows. The main reasons for not using these new technologies are reflected in social and technical factors (Alaloul *et al.*, 2020). So, understanding these technologies and trying to implement them in construction projects may help to improve the low efficiency of this industry and maximize the life cycle value.

#### 4.3. Digital Twin ecosystem and platforms

Cyber-physical system (CPS) is a term in the industry 4 revolution which provides a root for a bidirectional connection of physical assets and a virtual model. A lot of researchers and industrial parties recommend the integration of CPS with the cloud system to improve system efficiency, scalability, and communication ability. To provide the bidirectional connections, IoTs play a crucial role to transfer the data from the real world to the digital model or commands from the digital model to the real world. So, the cloud, physical assets, virtual model, and IoTs are the main parts of the DT model (Figure 15).

Currently, there are several solutions offered by companies that are widely used in different industries as a digital twin platform. General Electric, Microsoft, Siemens, Bentley, Autodesk, IBM, Cisco, and Oracle are some of the pioneer companies in the DT world. The number of active companies in the IoT and DT field is on the rise due to its importance and market demand. They all try to connect the physical and digital models with different services and analysis tools.



**Figure 15 - Digital twin ecosystem**

## 5. INTERNET OF THINGS (IoT)

### 5.1. Introduction

Due to the vast benefits of the IoT, it draws the attention of researchers, institutions, industry parties, and different stockholders in different fields. This technology is aligned with the technology of the internet which was introduced in the 1980s. With the IoT, we are moving towards a phase where all items in our environment will be connected to the Internet and will have the ability to communicate with each other with minimum human efforts. (Atlam, Walters and Wills, 2018). IoT can provide an occasion for Human-to-Human (H2H), Machin-to-Machin (M2M) and Human-to-Machin (H2M) interaction. The term things refer to everything around us from a small atom to a large ship. The IoT environment would allow users to manage and optimize electronic and electrical equipment's using the internet. (Suresh *et al.*, 2014). Although the IoT can cause some problems in the case of security and cybercrime (Atlam *et al.*, 2020), they should be managed and problems should be solved due to achieving huge gains in this field.

Recently, a lot of efforts have done in applying IoT in the appliance, even on ordinary ones. For instance, transferring conventional home appliances to IoT-enabled smart systems is done by Aheleroft *et al.* in 2020. They selected a conventional refrigerator and installed IoT sensors to check electricity consumption and temperature. Regarding that, they used WIFI as a communication tool and Ubidots as a platform for integration, monitoring, and analyzing information. (Aheleroft *et al.*, 2020). In another study, Gao *et al.* in 2021 tried to gather real-time data from the built environment. They prepared the low-cost framework for data acquisition through the connection of sensors and low-cost computers (Raspberry Pi 4 Model B) to aggregate data from sensors. They also made the database for integrating and managing those data in the central database server and applied their model to the real project and used different sensors such as temperature and humidity, Photodiode light, sound sensor, vibration, motion detection, combustible gas sensor, and carbon monoxide gas (Xinghua *et al.*, 2021).

Similar to the discussion in the DT section, IoT can be used in different fields. Some of these applications are shown in Figure 16 (Atlam, Walters and Wills, 2018).

For bettering understanding of this technology, it is necessary to know the IoT ecosystems and their connection method and also the classification of them which will be discussed in the following sections.

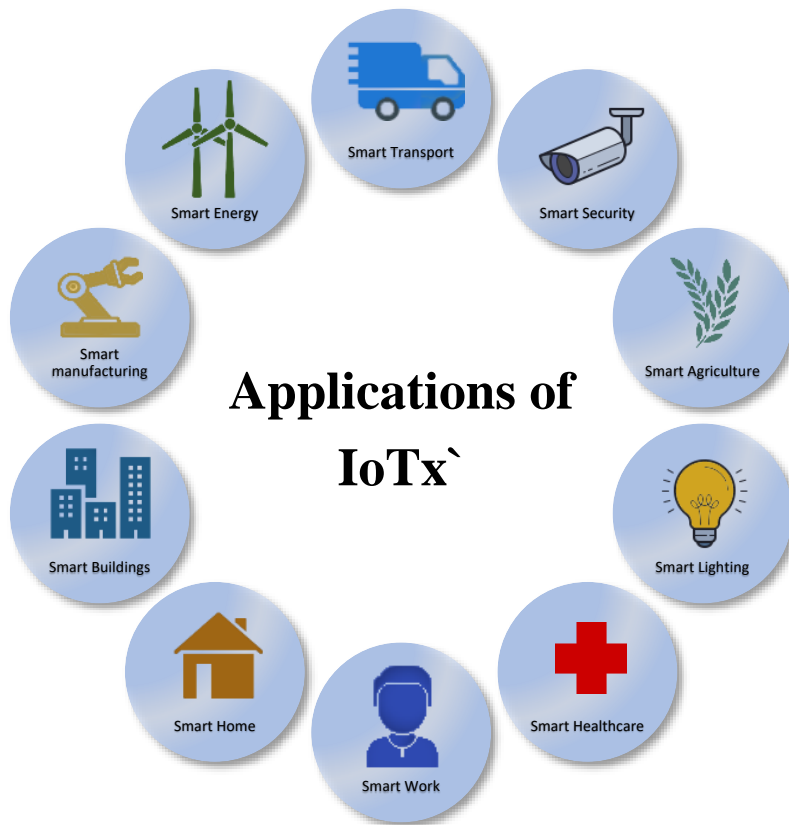


Figure 16 - Some IoT applications

## 5.2. IoT ecosystem and protocols

As we discussed, real-time data is an undetachable part of DT. So, understanding the IoT ecosystem plays an important role in constructing the DT model. A basic abstraction of how the IoT components connect is displayed by Costin et al. in Figure 17, which represents the data flow of smart objects through the IoT framework. (Aaron and Charles, 2019)

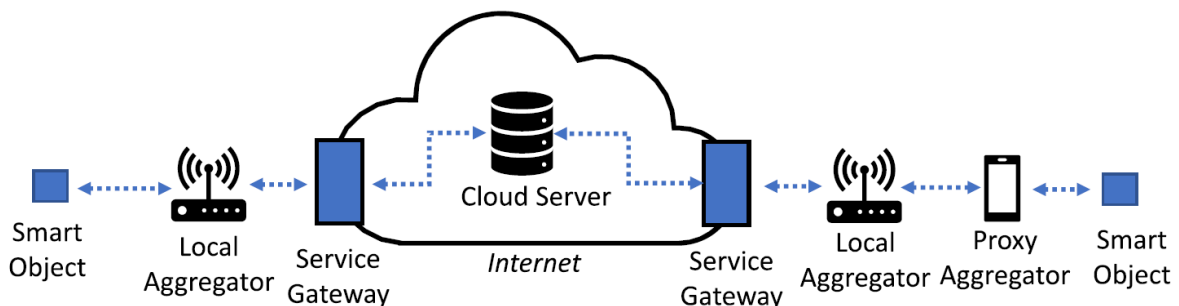


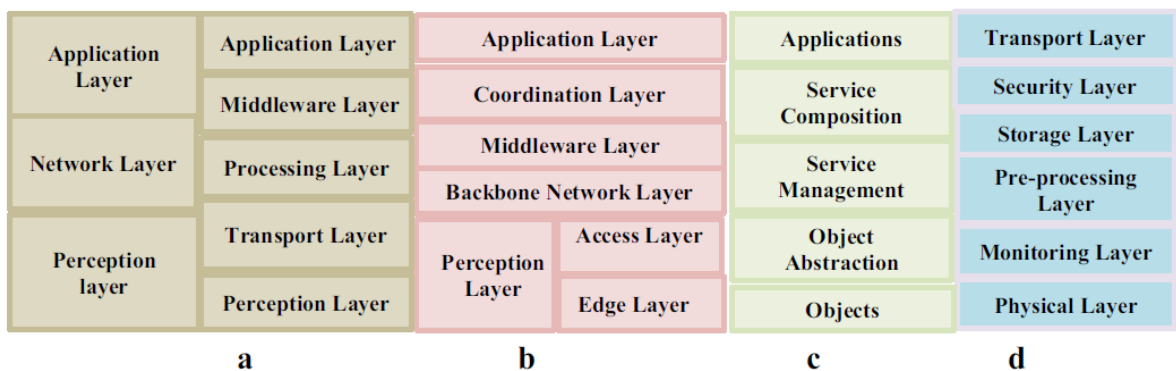
Figure 17 - Abstraction of the IoT components

To provide such a workflow, we need to know how the data can pass from the sensors to the applications or the cloud servers as the customers. Without communication, the flow of data will be disconnected

and we can not access the data in the final model. To have the same perception from different signals by IoT devices, protocols and standards have been created which can provide the possibility of better communication. Indeed, this is the concept of interoperability between IoT devices and networks which is one of the biggest challenges in this field.

Like BIM interoperability, there are a lot of efforts have been done in this regard. However, there are still lots of unmatched protocols in this field and also the number of protocols is on the rise. The diversity of protocols is normally because of the protocols’ different uses, similar to the situation that we cannot select the best option among different car manufactures or even the best car among one company’s cars. In addition, some other factors such as geography, trying to create a more efficient protocol that leads to having different protocols.

Although having different protocols and trying to create the more efficient one can improve the system capability, it can cause interoperability problems. So, understanding the protocol mechanism and compatible system is an important task. Accordingly, we have to have a good understanding of IoT layered structure. Based on each layer's role, we can find different protocols. Moreover, different IoT architectures can be developed based on different parameters such as security, range, etc. Some of these architectures are introduced by Bansal et al. in 2020 (Figure 18) (Bansal and Kumar, 2020).



**Figure 18 - IoT architectures, a) layered architecture—three and five-layer; b) middleware based; c) service-oriented based; d) fog based**

Although the three-layer model may not appropriately meet the security problem (Tournier *et al.*, 2020), this type is very common and is the basis for the other models. Each layer in this model has its features and protocols that cover this area. In summary, we can see this structure and some of the related protocols in Figure 19 which were proposed by Yugha et al. (Yugha and Chithra, 2020)

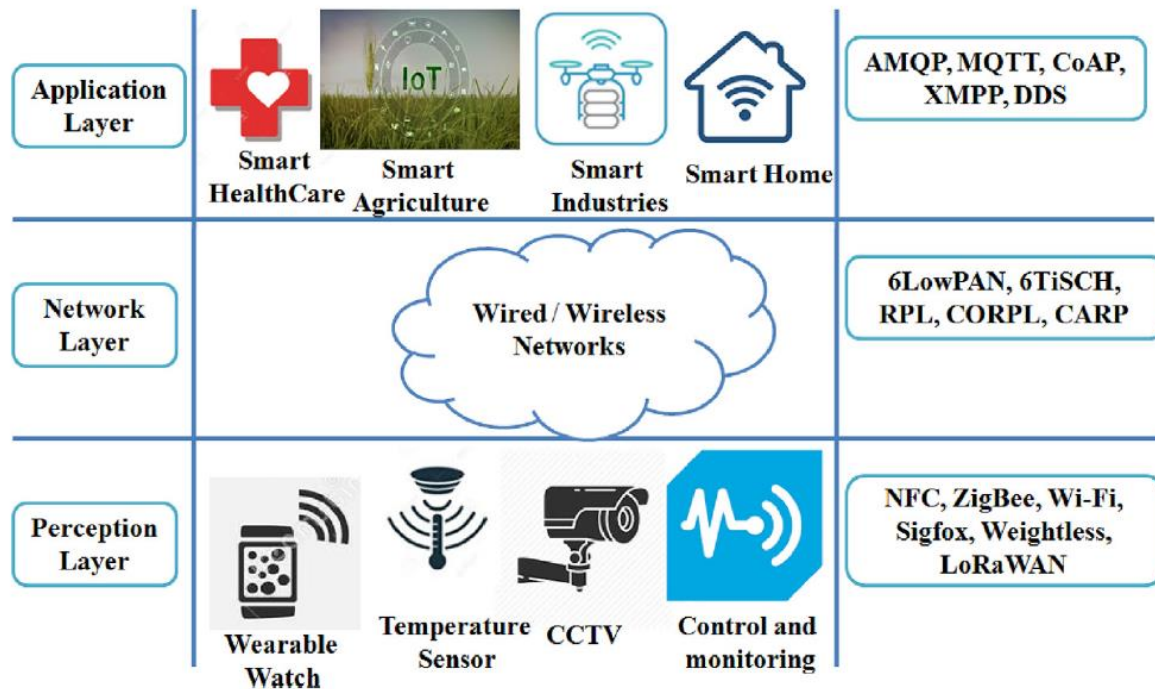


Figure 19 - Three-layer IoT architecture

### 5.2.1. Perception layer

This layer deals with the sensors and actuators that gather information and transmit them to the other layers. This information can be related to different criteria such as time, acceleration, humidity, Co2, etc. Besides, through this layer, users can send a signal back to the actuator to perform a task in the environment and connected devices.

Different technologies and protocols cover this layer such as NFC, Zigbee, Wi-Fi, Sigfox, Weightless, and Long Range Wide Area Networks (LoRaWAN) which connects the physical environment to the network (Yugha and Chithra, 2020).

#### Near Field Communication (NFC)

Near Field Communication (NFC) is a protocol that provides a wireless connection between devices at a very short range (up to 4 Cm) (Atlam, Walters and Wills, 2018) (Yugha and Chithra, 2020). Although this short-range provides limits in connection, this connection by default has higher security. NFC operates in a frequency range of 13.56 MHz and has a transmission rate of up to 0.4 Mbps (Coskun, Ozdenizci and Ok, 2013). Normally, NFC technology is common in contactless payments via mobile, wallets, etc. it also can be used for authentication and access control, data transfer, ticketing.

NFC has two types: active and passive. While in active type, both devices generate a magnetic field, magnetic fields just being produced by one connected device in passive one. (Bansal and Kumar, 2020). These types are shown in Table 3 (Coskun, Ozdenizci and Ok, 2013).

**Table 3 - Active and passive communication mode**

Device A	Device B	RF Field generation	Communication mode
Active	Active	Generated by both devices	Active mode
Active	Passive	Generated by Device A only	Passive mode
Passive	Active	Generated by Device B only	Passive mode

### ZigBee

Zigbee, a network protocol developed by Zigbee alliance is an IEEE802.15.4 standard. This protocol which provides a range of up to 150 m in outdoor places, needs a very small power. Since the bandwidth that it uses is different from WIFI and Bluetooth, interferences can not occur among them. It can transfer data up to 250 kbps. (Ramya, Shanmugaraj and Prabakaran, 2011).

Although this technology is cheap and compatible with different network topologies such as mesh, star, and tree, it is suitable for applications that need a low data rate (Atlam, Walters and Wills, 2018)

### Wi-Fi

Wireless-Fidelity (Wi-Fi), one of the wireless connection protocols, is one of the IEEE 802.11 families standards. Wi-Fi is an alternative network to a wired network that is commonly used for connecting different devices such as tablets, mobiles, computers, etc. for transferring data. Wi-Fi, which is categorized in short-range networks, has a transfer data rate up to 320 Mbps and covers up to 300 m (Zemrane, Baddi and Hasbi, 2018).

### Sigfox

This protocol provides wireless connections for devices in long ranges. Sigfox range is up to 50 km in rural environments and runs in a small battery. While it is cheaper than Wi-Fi and has longer coverage, its data transfer rate is 10 to 1000 bps (Ray and Bagwari, 2017).

### Weightless

Weightless is the name of a set of technology for Low-Power Wide-Area Network (LPWAN). These technologies are managed and developed by Weightless Special Interest Group (Weightless SIG). Weightless-P, Weightless-N, and Weightless-W are some of its variations (Oliveira *et al.*, 2019). This technology can be used in traffic sensing, rural broadband communication systems, industrial machine monitoring systems, smart e-payment, and smart appliances management systems with a covered area up to 10 km (Yugha and Chithra, 2020).

## **LoRaWAN**

Long-Range Wireless Area Network (LoRaWAN) is similar to Sigfox. The typical coverage range is 5 km and 15 km for urban environments and rural environments, respectively. (Ray and Bagwari, 2017). The data transfer rate in this protocol is up to 50 kbps (Tournier *et al.*, 2020)

### **5.2.2. Network layer**

This layer is responsible for processing and transmitting the received data. They try to find the optimum path to transfer the data between network hosts. This layer uses (IPsec) protocol which provides end-to-end security for authentication, confidentiality, and integrity. IPsec security is mandatory in Ipv6 (Internet Protocol version 6) for initiating the services among all applications. The protocols used in the network layer are 6LowPAN, 6TiSCH, RPL, CORPL, and CARP. (Yugha and Chithra, 2020).

## **6LowPAN**

IPv6 low power wireless personal area networks (6LowPAN) is an IP-based technology that is designed to transmit Internet Protocol version 6 (IPv6) packets. This protocol was developed in 2007 with the concept that devices with limited processing capabilities should be able to use the IP protocol and participate in the Internet of Things. The signal range is up to 100 m and the maximum data rate is 250 kbp (Hofer-Schmitz and Stojanović, 2020).

## **RPL**

Routine Protocol (RPL) is the standard protocol by creating a directed graph for all the nodes in the network. This protocol defines how the data would be passed from one device to another and also to the gateway (Bansal and Kumar, 2020).

## **CORPL**

Cognitive and Opportunistic RPL (CORPL) is the extension of RPL and is specifically designed for Cognitive radio networks. RPL protocol has been extended with the support of an opportunistic forwarding approach (Yugha and Chithra, 2020). It uses opportunistic forwarding to forward a packet between the nodes. Each node of CORPL protocol keeps the information of forwarding set rather than parents that only maintaining it.

## **CARP**

Channel-Aware Routing Protocol (CARP) is a lightweight multi-hop delivery of data to the sink which is used in the IoT network for data forwarding and network initialization. It does not store any history of data which is not suitable for IoT applications where data is highly dynamic (Yugha and Chithra, 2020).

## **6TiSCH**

This protocol is aimed at standardizing the mechanism of the industrial internet of things as a reality and also a new version, 6TiSCH IETF was released in 2013. This protocol envisions several small wireless

mesh networks federated by a spanning high-speed backbone (Dujovne *et al.*, 2014). This protocol which is now under development focuses on transferring data packs from data source to destination in the network layer.

### **5.2.3. Application layer**

The application layer provides the information for end-user and customers. This layer delivers the requested information like humidity, Co2 index value, or any other data that was supposed to be transferred. Some of the protocols that cover this layer are AMQP, MQTT, COAP, XMPP, and DDS. (Yugha and Chithra, 2020).

#### **AMQP**

Advanced Message Queuing Protocol (AMQP) is an effort for message exchange and communication. This protocol is highly scalable and is being used in business and commercial platforms. Although it is highly extendable and interoperable in different platforms, it is not suitable for the constrained environment and real-time applications and does not support an automation discovery mechanism (Yassein, Shatnawi and Al-zoubi, 2016).

#### **MQTT**

Message Queue Telemetry Transport (MQTT) is a lightweight and open-source protocol that was designed in 2013 by IBM. Minimizing the network traffic and lower latency are the most features of this protocol. (Yassein, Shatnawi and Al-zoubi, 2016). To improve its security, the Secure MQTT (SMQTT) version was also released (Yugha and Chithra, 2020).

#### **COAP**

“Constrained Application Protocol (CoAP) is request/response protocol; it is similar to a client-server model. Nevertheless, this protocol is only sufficient in a constrained environment such as a constrained node with low capability in RAM or CPU and constrained network, such as lower power using Wireless Personal Area Network (WPAN). This constrained environment led to bad packet delivery and high overhead. CoAP was designed by the Internet Engineering Task Force (IETF) which is mainly interested in the machine to machine (m2m) applications and the automation of systems to reduce overhead, enhance packet delivery, and increase the simplicity of work, by using a simple interface with HTTP” (Yassein, Shatnawi and Al-zoubi, 2016)

#### **XMPP**

Extensible Messaging and Presence Protocol (XMPP) is a common communication protocol in the IoT ecosystem. This protocol is based on Extensible Markup Language (XML) that supports both request/response and publish/subscribe models which allows bidirectional communications and multi-directional communication. In summary, this protocol is simple and flexible, however, it needs high consumption bandwidth and high CPU usage (Yassein, Shatnawi and Al-zoubi, 2016).

## DDS

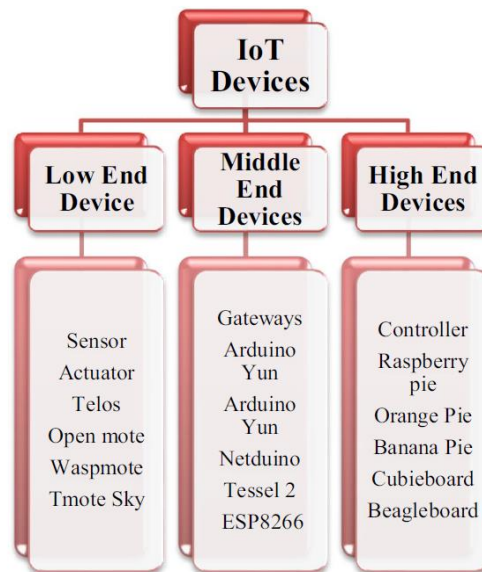
Data Distribution Service (DDS) is a protocol suitable for real-time applications. DDS is now widely used in warship control, flight simulator, and military network system. (He and Liang, 2015). So, it can be used for low latency IoT devices with high reliability (Yugha and Chithra, 2020).

### 5.3. IoT devices

Generally, IoTs can be investigated based on their properties. A thorough study on the devices based on these features can lead to a better selection of desired IoT. Some of these properties are as follows (Jincy and Sundararajan, 2015):

- Power Constrained or Non-Power Constrained
- Real-Time or Non-Real Time
- Closed Loop and Open Loop
- Communication Protocol Used
- Wired and Wireless Systems
- Network Size
- Interoperability
- Reliability
- Energy Source.
- Processing Speed
- Data Management
- Memory
- Scalability
- Bandwidth
- Size of Sensors
- Functionality

Generally, as depicted in Figure 20, IoT devices are categorized into 3 levels (Bansal and Kumar, 2020):



**Figure 20 - Categorization of IoT devices**

#### Class 0 or low-end IoT devices

- Devices with limited resources like memory, power, computational capability, architecture, etc.
- Mostly present at the first layer of the IoT ecosystem.
- The functionality of sensing and actuating
- They have a very basic operating system.
- RAM varies from 1 to 50 KB and flash memory varies from 10 to 50 KB.
- These devices are very much vulnerable to threats and security is the biggest concern in these low-end devices.
- These devices are closer to environmental parameters like temperature, humidity, pressure, etc.

#### Class 1 or middle-end IoT devices are

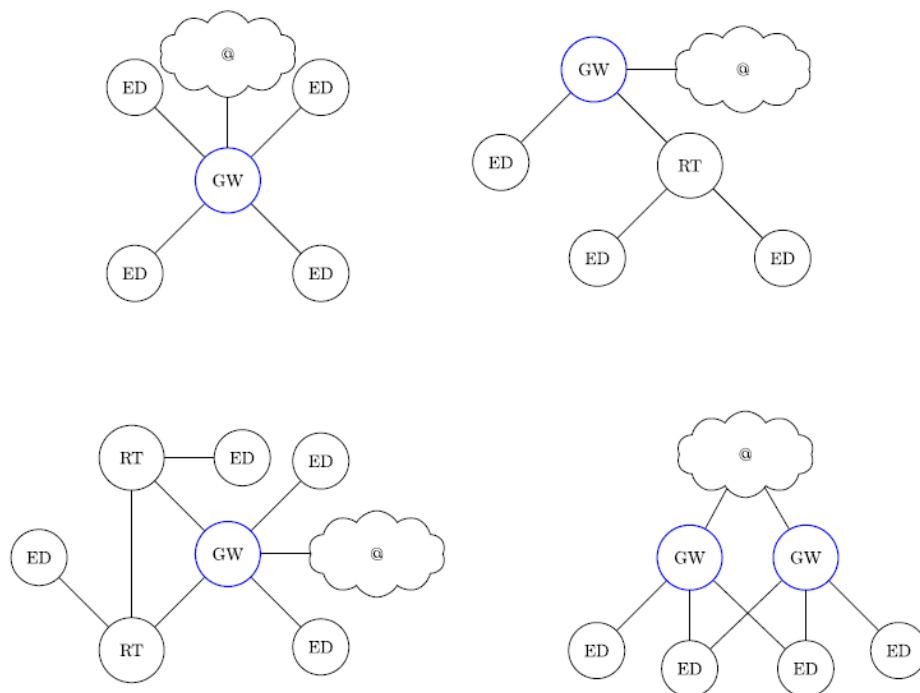
- Devices that have more resources as compared to low-end devices.
- They provide more functionality than low-end IoT devices but the computational capability is not enough to handle very complex requirements.
- These are basic microcontrollers.
- These devices sit over low-end devices in the IoT ecosystem to manage and improvise the capabilities of low-end IoT devices.
- These devices have capabilities like image processing, data

- The RAM varies from 100 KB to 100 MB and flash memory varies from 10 KB to 100 MB

Class 2 or high-end IoT devices

- They are single-board computers that have a high number of resources in terms of CPU, RAM, flash memory, etc.
- These devices support traditional operating systems such as LINUX, UNIX.
- These devices support growing technologies like artificial intelligence, machine learning, deep learning, nature language processing, etc. These devices also provide a graphical user interface for development purposes. These devices are available for up to 64-bit architectures.
- These devices have everything on a single board and support almost all communication protocols.
- The devices have many connectivity interfaces such as HDMI, Wi-Fi, Ethernet, USB, Bluetooth, etc. and all are present on a single board.
- These are highly intelligent devices used as complex gateways and controllers in the IoT ecosystem.
- These have relatively fewer security concerns due to the high number of resources.

Based on the IoT properties and classes, they can play a special role in the IoT network. The architecture of the network can be varied, however, some of these topologies are depicted in Figure 21 (Tournier *et al.*, 2020).



**Figure 21 - Types of topology: gateway (GW), router (RT), and end device (ED)**

In the above topologies, which are named star, tree, peer to peer (P2P), and cellular, respectively, we can see 3 different nodes: ED, RT, and GW.

In network topologies, we hear the name of some devices such as Router, switch, hub, etc. here we describe more these devices for more clarification

### **Repeater**

Repeaters are devices that are in charge of amplifying the received signals for transferring to a longer distance. It is noteworthy that they do not amplify the signals, they just copy the signal bit by bit when they are weak.

### **Hub**

These electronic devices are responsible for connecting multiple computers in the network. They also play the role of repeaters for strengthening the signals. It should be noted that they can only connect LAN components with the same protocols. Besides, they cannot filter data and are not intelligent enough to find the best path for transferring data.

### **Bridge**

These devices are repeaters with the ability of filtering by reading the Media Address Control (MAC) addresses. Bridges are like hubs with the ability of filtering. Like hubs, they can connect components with identical protocols.

### **Switch**

These devices act like hubs. However, with their intelligence, they can improve network efficiency by error-checking and forward messages selectively to the correct port only.

### **Router**

Routers are devices like switches. In addition, they can connect Local Area Networks (LAN) and Wide Area Networks (WAN) together. It means that can work with different protocols of WAN and LAN.

### **Gateway**

As it can perceive from these devices, they act as a gate and connect different networks that work upon different networking models. They play the role of hubs and switches in a more complex way and they are capable of converting protocols.

This page is intentionally left blank

## 6. CLOUD COMPUTING

### 6.1. Introduction

As we explained in the industry 4 section, cloud computing or cloud is the fast-growing technology that has attracted researchers' attention. Due to its beneficial services, using the cloud in different fields is a new trend. Cloud computing "is a model for allowing ubiquitous, convenient, and on-demand network access to many configured computing resources (e.g., networks, server, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Hashem *et al.*, 2015).

Cloud is emerged to offer a wide range of services, including servers, storage, databases, networking, software, analytics, and intelligence on the internet. Cloud services models normally consist of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (Hashem *et al.*, 2015) (Bittencourt *et al.*, 2018). By the passage of time, other services such as Function as a Service (FaaS) is introduced which is one more step in the application layer that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app (Aslanpour, Gill and Toosi, 2020)(Aceto, Persico and Pescapé, 2020). It is fundamental for enterprises to select the right cloud solution to address their challenges and have the proper investment in cloud services.

Another cloud classification is based on the rights that make deployment models (Bittencourt *et al.*, 2018):

#### **Public cloud**

These clouds that may be owned, managed, and operated by a business, academic, government, or combination of them are being provided for general public uses.

#### **Private cloud**

Private clouds are provisioned for exclusive use by a single organization for example business units. It may be owned, managed, and operated by the organization, a third party, or some combination of them.

#### **Community cloud**

The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them.

#### **Hybrid cloud**

Hybrid clouds are a combination of two or types of mentioned models defining the rules and standards.

Generally, cloud computing can be beneficial in the case of cost, work scale, performance, productivity, etc. However, due to the big data produced by IoT devices, there are some challenges related to the data transferring between the end-user and cloud system such as latency, bandwidth, cost, and availability of connection (Caiza *et al.*, 2020). Regarding that, fog computing is deployed to improve the network capacity for delivering edge services.

The following sections will discuss PaaS, SaaS, IaaS, and fog and edge computing for more clarification.

## 6.2. IaaS

As mentioned earlier, there are three main services offered by cloud servers. All these services can be beneficial for the customers in case they be selected wisely based on users' needs. Figure 22 which is offered by Kavis (Kavis, 2014) shows their differences. For comparing these services with the traditional one, in traditional one the customer should think about all the processes from providing the physical hardware, installing the operating systems, firmware, applications, and so forth. Also, the operation and providing security is another aspect that should be noted. As it is depicted in this schema, each service reduces the efforts that are required by the customer.

In IaaS, the server delivers computing, storage, networking, and other primary computer services, and the customer can install and manage virtual machines, operating systems, and applications. So, the user does not care about the infrastructure and has control over the operating system and applications (Mustafa and Zeebaree, 2021). Today. There are a lot of companies that offer IaaS services such as Amazon Elastic Compute Cloud (EC2) and S3, Nirvanix, IBM Bluehouse, Linode, OpenStack, GoGrid, and Rackspace cloud (Mustafa and Zeebaree, 2021) (Kavis, 2014) (Shallal and Bokhari, 2019).

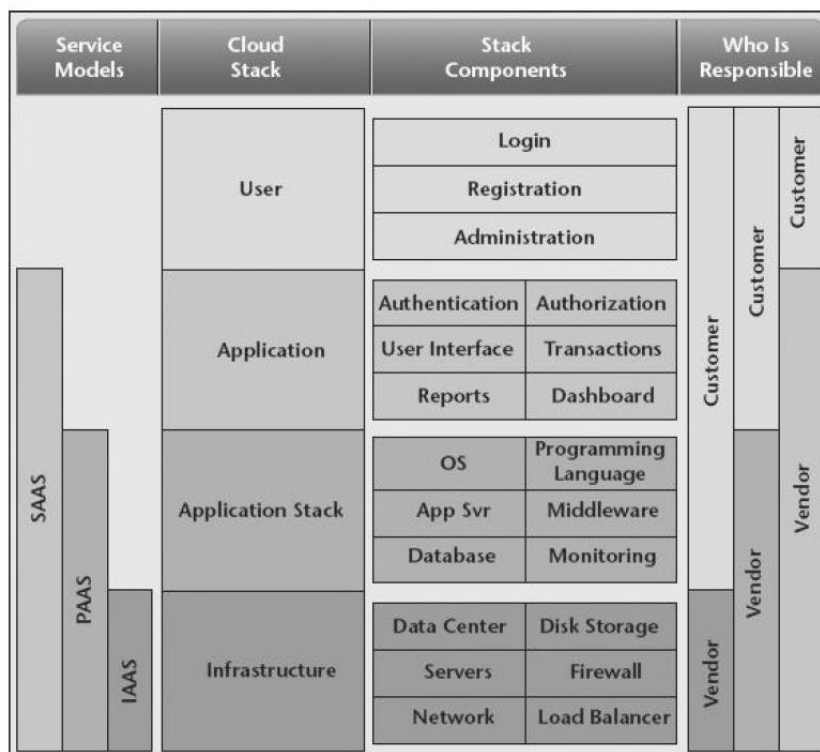


Figure 22 - Cloud Stack

### 6.3. PaaS

The next step over IaaS is PaaS. In this service, PaaS provides services including Operating Service (OS), programming language, database, a web server that automatically scales to meet the users (Naren.J, Sowmya and Deepika, 2014). So, the consumers are not in charge of managing or controlling underlying cloud infrastructures such as networks, servers, OS, or storage. Hence, they focus on applications and application-hosting environments. Vendors such as Force, Google Apps Engine, Microsoft Azure, Salesforce provide this service (Naren.J, Sowmya and Deepika, 2014)(Bittencourt *et al.*, 2018)(Dillon, Wu and Chang, 2010)(Kavis, 2014)

### 6.4. SaaS

SaaS is the most basic service and it is the top layer in cloud services. This service provides an opportunity for the consumers to use the applications offered by providers. Web browser or a program interface are the ways that consumers have access to this service through them (Mustafa and Zeebaree, 2021) (Kavis, 2014). Customer Relationship Management (CRM), Enterprise Resource Planning (ERP) payroll, accounting, and other uses are common applications in this field.

To summarize and a general comparison between these services, we can see their features in Table 4 which is provided by Bokhari et al. (Shallal and Bokhari, 2019).

**Table 4 - Comparison between SaaS, PaaS, and IaaS**

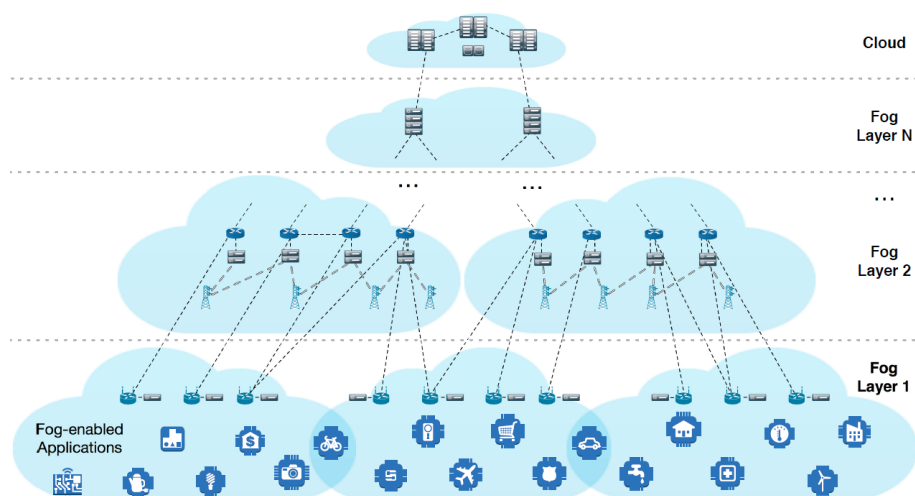
Classification	Service type	User control on	Provider control on	Flexibility/generality	Difficulty level	Scale	Level	Vendors
<b>SaaS</b>	Application with specific function	No thing	Application, data, runtime, middleware, O/S, virtualizations, servers, storage, network	Low	Easy	Small	User level	SalesForce.com, Google documents, Facebook.com, Gmail, Hotmail, Quicken Online, IBM, WebSphere, Oracle on-demand, SAP, Netsuite
<b>PaaS</b>	Application hosting environment	Application, data	Runtime, middleware, O/S, virtualizations, servers, storage, network	Middle	Middle	Middle	Developer level	Google AppEngine, Microsoft Azure, Manjrasoft Aneka, Coghead, Force.com, Yahoo Developer Network, MSFT, Heroku, Mosso, Engine Yard
<b>IaaS</b>	Basic computing, storage, network resource	Application, data, runtime, middleware, O/S	Virtualizations, servers, storage, network	High	Difficult	Large	IT level	Amazon EC2 and S3, Nirvanix, Op source, GoGrid, RACKSPACE, IBM BlueHouse, Linode, VMWare

## 6.5. Fog and edge computing

As we talked about IoT systems, Cloud computing is one of the fundamental pillars in its ecosystem. Based on the main goal of the IoT systems, connecting all the devices and things to the internet, the number of connected devices and data in the network is on the rise. It is predicted that the number of connected IoT devices will reach 70 billion by 2025 (Liu, Wang and Zhang, 2020). This huge amount of data produced by connected devices cause difficulties and affect the Quality of Service (QoS). Bandwidth, latency, connectivity, privacy, cost, and energy consumption are some of the criteria that can be affected by this big data.

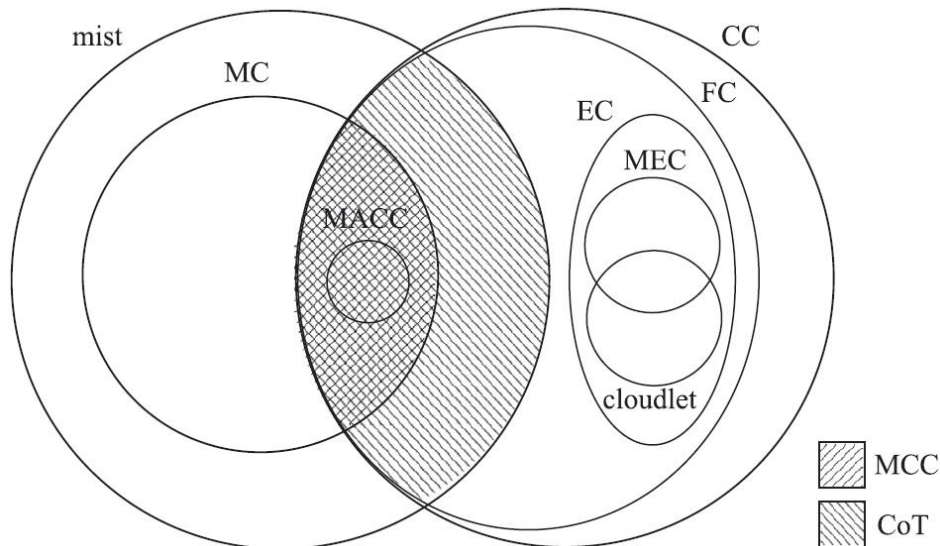
To overcome limitations, different techniques are being developed. Edge computing and fog computing are two of these efforts to improve QoS. Especially in latency-sensitive applications such as health monitoring or some industrial one, latency in sending and receiving data is unacceptable (Dastjerdi and Buyya, 2016). Based on Yi definition, “fog computing is a geographically distributed computing architecture with a resource pool consists of one or more ubiquitously connected heterogeneous devices (including edge devices) at the edge of the network and not exclusively seamlessly backed by cloud services, to collaboratively provide elastic computation, storage and communication (and many other new services and tasks) in isolated environments to a large scale of clients in proximity.” (Yi et al., 2015)

Fog computing is based on the principle to bridge the IoT to the cloud by enabling computing, storage, networking, and data management within the close vicinity of IoT devices at the edge of the network (Yousefpour et al., 2019). The data transfer rate and computation demand will be reduced with the presence of these edge devices by handling some of these tasks locally without referring to the cloud. The number of fog nodes is more than the clouds and they are geographically available in large numbers. Indeed, these techniques try to improve connection scalability. Generally, scalability is a feature of a system about the quality of system services and system capabilities under a large number of demands. These intermediate devices make new layers between IoT devices and the cloud. Thanks to technological progress, even the hierarchical fog paradigm is doable which is shown in Figure 23 by Bittencourt et al. (Bittencourt et al., 2018).



**Figure 23 - IoT-Fog-Cloud connectivity**

It is noteworthy that cloud computing and fog computing are complementary and one can not replace the need of the other (Yousefpour *et al.*, 2019). Based on the mentioned definition and the aim of technologies in this field, we have different computing paradigm depicted by Yousefpour et al. in Figure 24, and the paradigm acronyms that are shown in Table 5 (Yousefpour *et al.*, 2019).



**Figure 24 - A classification of the scope of fog computing and its related computing paradigms. (Intersection of cloud computing and mist computing is Cloud of Things, and the intersection of mobile computing and cloud computing is mobile cloud computing.)**

**Table 5 - A list of computing paradigm acronyms used in this section**

IoT	Internet of Things	CC	Cloud Computing
MC	Mobile Computing	FC	Fog Computing
EC	Edge Computing	MEC	Multi-access Edge Computing
MCC	Mobile Cloud Computing	MAACC	Mobile ad hoc Cloud Computing
CoT	Cloud of Things	mist	Mist Computing

### Mobile Computing (MC)

Thanks to technological progress in mobile devices in general and smartphone in particular, they can play a role in the network environment. These devices provide an opportunity to access electronic mail, sending faxes, accessing the web or remote databases, and cellular telephones or local networks even when people are in travel (Chalmers and Sloman, 1999). Generally, mobile computing, which the mobile device can be a tablet, laptop, or smartphone, is when part of the computation and data preparation is

done by these mobile devices. Their access to different sensors such as accelerometer, Global Positioning System (GPS), gyroscope, camera, microphone, and a proximity sensor made them flexible and useful computing devices as part of the network environment.

### **Mobile Cloud Computing (MCC)**

Mobile Cloud Computing (MCC) is defined as an infrastructure where both data storage and data processing occur outside of the mobile device (Yousefpour *et al.*, 2019) (Yi *et al.*, 2015). Delegation of the storage and computation tasks to the remote entities done to optimization of energy consumption, lifetime, and cost. This idea also brings mobile devices to a wider range of users even without powerful mobile devices (Donno, Tange and Dragoni, 2019)

### **Mobile ad hoc Cloud computing (MACC)**

Mobile ad hoc Cloud Computing (MACC) is a paradigm that is different from MCC. In this scenario, mobile devices must join continuously the network. This form of decentralized paradigm consists of nodes that form a temporary, dynamic network through routing and transport protocols (Yousefpour *et al.*, 2019).

### **Edge Computing (EC)**

Similar to the MCC, this paradigm is based on the assumption to delegate the storage and computation tasks on other devices and nodes. However, in EC, the computing nodes are at the edge of the network, close to the IoT devices but not located on them. Based on Shi definition, edge computing “refers to the enabling technologies that allow computation to be performed at the network edge so that computing happens near data sources.” (Shi and Dustdar, 2016)

### **Multi-access edge computing (MEC)**

European Telecommunications Standards Institute (ETSI), defined Multi-access Edge Computing (MEC) as a platform for leveraging edge computing in any radio access network such as mobile network and wireless network (Antony Franklin and Tambe, 2020). Video analytics, connected vehicles, health monitoring, and augmented reality are examples of MEC (Yousefpour *et al.*, 2019).

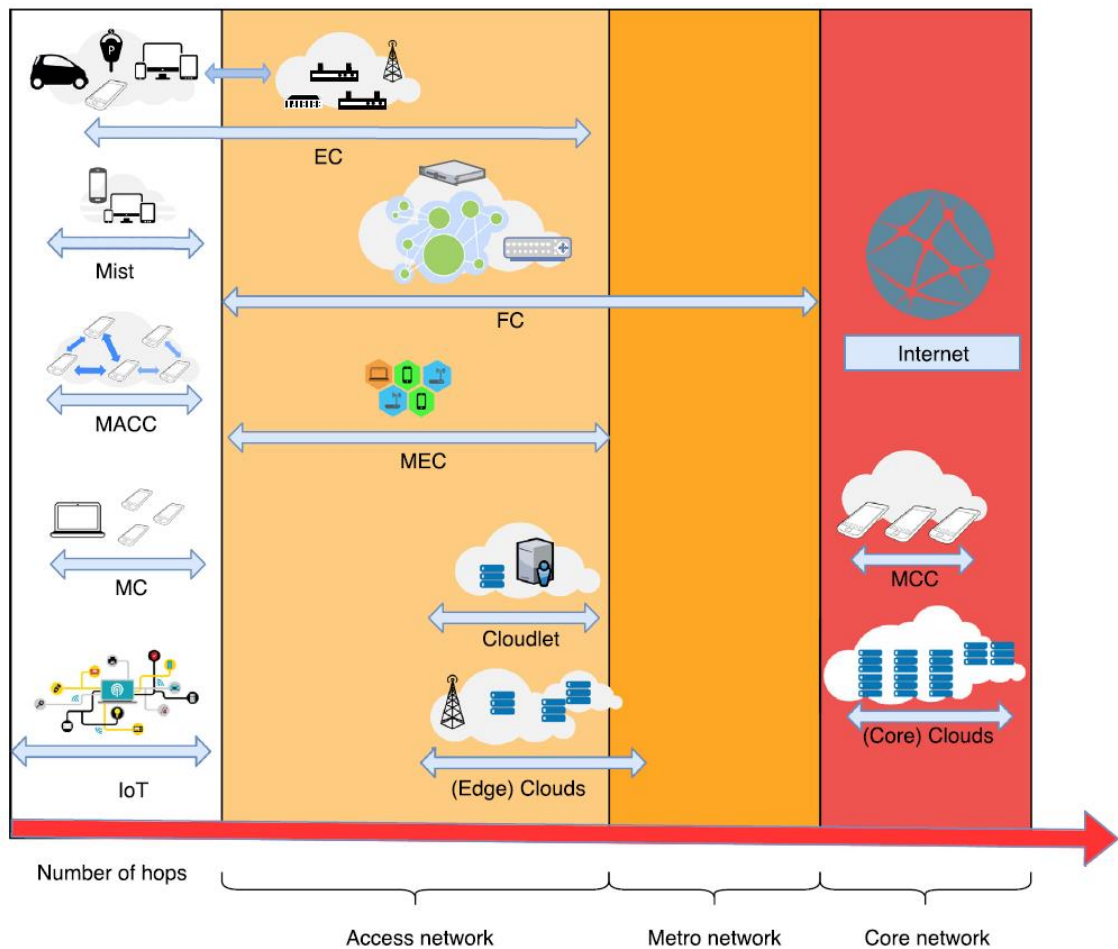
### **Cloudlet computing**

Cloudlets are trusted smart data centers which follow the cloud computing paradigm in a more concentrated manner and rely on high-volume servers. (Yi *et al.*, 2015) (Dolui and Datta, 2017). Although MAAC and cloudlet computing has similarity, MAAC is resource-constrained and lacks virtualization support for real-time IoT applications (Yousefpour *et al.*, 2019)

### **Mist computing (MIST)**

When IoT devices are computation-enabled and capable to cooperate, make the mist computing layer. RaspberryPi is an example of such devices with the ability of computation as well as typical functionality of devices such as sensing and actuating (Aslanpour, Gill and Toosi, 2020)

For comparing the fog computing paradigms, we can see their differences based on their distance from the core cloud in Figure 25 (Yousefpour *et al.*, 2019).



**Figure 25 - Comparison of fog computing and its related computing paradigms in terms of their location and distance from the core clouds.**

In summary, the general atmosphere of cloud and edge computing and their dynamic environment are described in this section. As mentioned, fog and edge computing is an undetachable part of the IoT environment. Regarding the network demands, some common paradigms ARE described as an example and there is a lot of other new paradigms that are under development to address new challenges.

This page is intentionally left blank

## 7. CASE STUDY

### 7.1. Introduction

Cascina Nosedo belongs to the type of Lombard agricultural farmhouse with a closed courtyard and is located in the extreme south-eastern outskirts of the city of Milan on the border between Municipio 4 and Municipio 5. The farmhouse is made up of two large distinct bodies, the most ancient dates back to the fourteenth century. The entire area has a size of 3,324 m<sup>2</sup> and today includes nine buildings, up to the nineties there were ten buildings (Figure 26).



**Figure 26 - Cascina Nosedo project**

Part of the property has been owned by the Municipality since July 2015 and tried to redevelop the internal areas to promote any future social cohesion projects such as social and work inclusion, recreational, artistic, and hospitality activities.

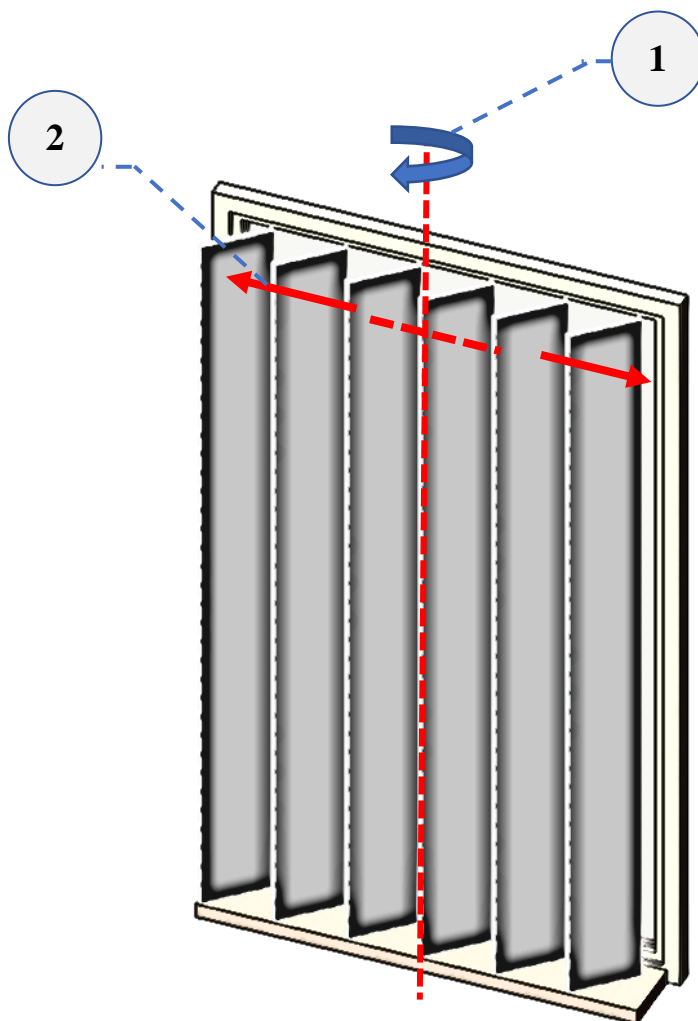
Cascina is a project which is supposed to be sustainable, smart, and efficient in case of environmental issues. Regarding being a smart building and reducing the amount of energy consumption, using daylighting can be helpful and provide a pleasant atmosphere for users.

Daylighting should be considered as the main lighting system which can be complemented by artificial lighting. This mechanism can be efficient as it can reduce energy consumption while providing a pleasant situation for occupants to meet their needs. Different parameters affect buildings' daylighting such as project location, building direction, spaces layout, openings, windows, etc. Since the Cascina building is under construction and the design phase has been finished, an innovative dynamic louver is designed within the DT model to control the openings manually or automatically by considering the sunlight direction at different times of the year.

This automated system provides an opportunity for defining different scenarios in different seasons and times of the year. For instance, if the indoor light is not enough in summer, the louvers can be opened

automatically while they are perpendicular to the sunlight direction to hinder the entrance of the direct sunlight. while this system provides sufficient indoor light, it meets the occupants' IEQ requirements and prevents light dazzling. On the other hand, in winter, the maximum sunlight entering is pleasant due to the energy consumption and occupants' comfort. So, the louvers should be able to rotate and locate parallel to the sunlight in winter. Moreover, the system should be able to control the louvers manually based on the user's request. In addition, having the indoor light status is useful for automation which should be derived in a real-time manner.

Based on the aforementioned criteria, the dynamic window is designed with movable louvers which can rotate around their vertical axes or go toward the sides, separately (Figure 27).



**Figure 27 - Window with movable louvers**

All these scenarios can be applied if we have access to the current light status and also the infrastructure for controlling the windows louvers. DT is a good solution in this project that can help us both getting the environmental information and sending the commands. Indeed, the DT model will be used here for visualization, simulation, and automation. In the following chapters, the creation of the DT model and the results will be described.

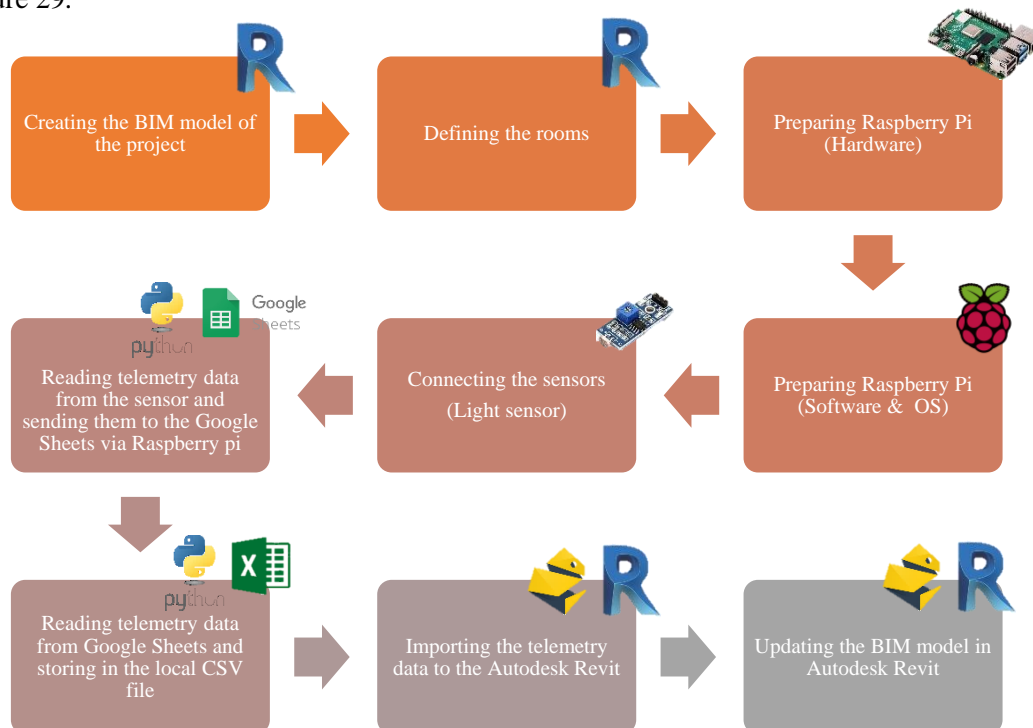
## 7.2. Methodology

Different workflows can be considered for providing the DT model. However, the imperative matter is providing the real-time (or near real-time) data from the environment and providing the situation for demonstrating them in a 3D model. Also, it is supposed that some commands could be transmitted from the model to the actuators in a physical model to control the facilities.

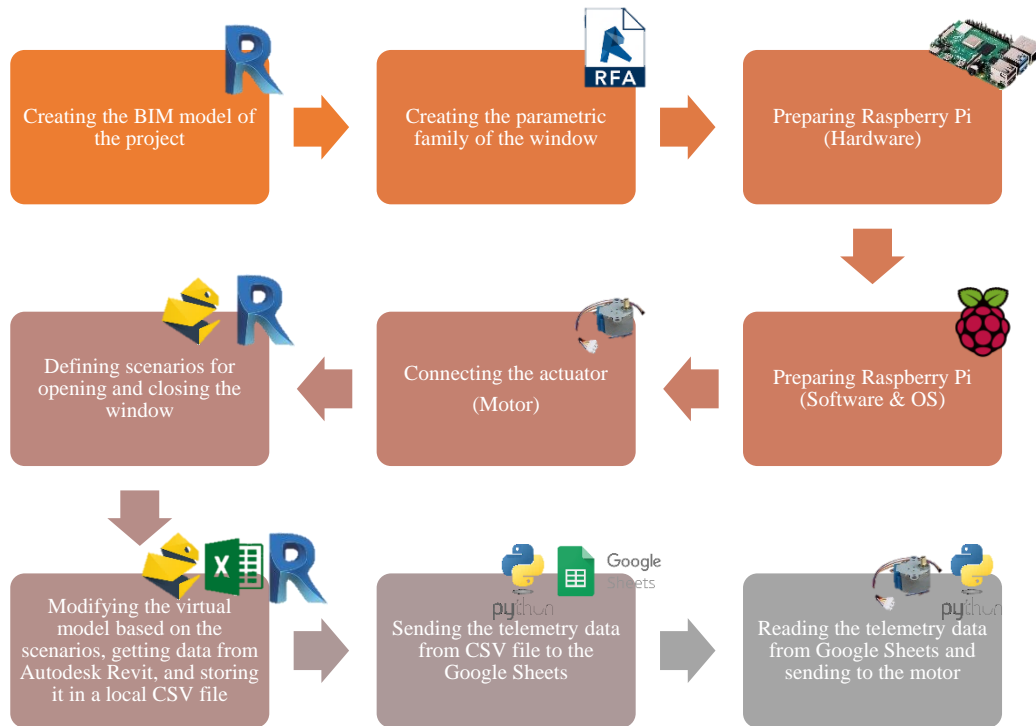
As one of the goals in this study, providing the low-cost DT model, the Raspberry Pi 4, a single-board computer, is used as a core for getting the environmental data and sending them to the cloud and BIM model. It also receives the commands from the BIM model through the internet and sends them to the motor and actuator. In addition, Autodesk Revit is used for visualization and integrating the real-time data and the 3D model. Noteworthy that Autodesk Revit software is selected as it is very common in the AEC industry and is a known authoring software, so there is no need to buy extra software. Also, a plugin is designed for Revit to make easy the data streaming between the virtual model and physical model with the Python and pyRevit scripting.

The whole process for creating the DT model and using it for automation and simulation can be divided into 2 parts: 1) collecting the environmental real-time data (light status) and sending them to the BIM model, 2) receiving the commands (louvers' rotation angle) from the BIM model and sending them to the motor and actuator.

These two parts require both hardware, software, and programming knowledge for integrating the data streaming. For a better understanding of the whole process, two workflows are depicted in Figure 28 and Figure 29.



**Figure 28 - The workflow of collecting the environmental data (light status) and updating the BIM model in a real-time manner**

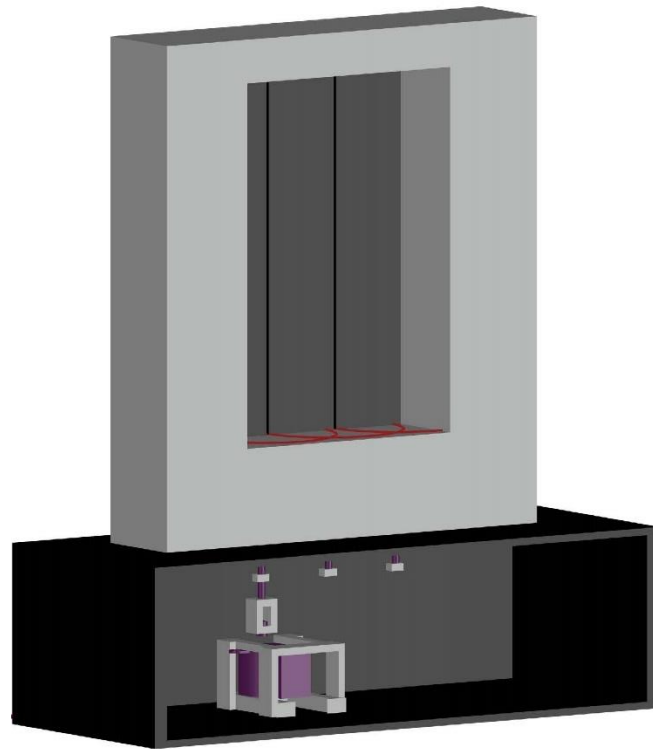


**Figure 29 - The workflow of getting the commands from the BIM model and sending them to the motor**

Selecting the Google Sheets, a SaaS cloud service for data storage, not only provides easy access to the data and monitoring but also enables the user to control the window easily via just a smartphone or tablet without installing extra software.

These two workflows can be merged at some parts such as receiving and sending the data in Raspberry Pi, or receiving and storing the data from Google sheets to the local CSV file and vice versa. However, for more clarification and description, they are described separately in this context.

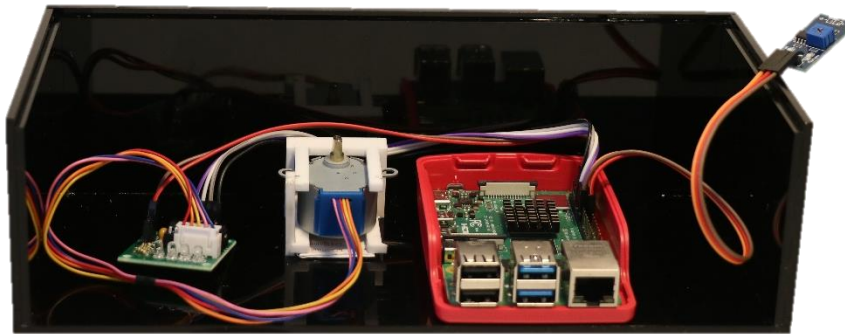
Since the project is still under construction, the physical prototype of the designed window is constructed for testing the process. Regarding that, for a more precise model, first, the virtual model is designed in AutoCAD 3D (Figure 30), and then it was constructed based on the model (Figure 31). In this physical model, the louvers can rotate based on the received commands which are exactly similar to the BIM model rotation. For rotating the louvers, the stepper motor, and Raspberry Pi board are located on the bottom of the model to rotate the lovers (Figure 32). Moreover, the light sensor is embedded that can reflect environmental light status to the BIM model. These functionalities and the process will be explained in the following parts.



**Figure 30 - The 3D model of the physical model**



**Figure 31 - The physical prototype of the project's windows**



**Figure 32 - The embedded motor, boards, and light sensor in the constructed physical model**

### 7.3. Collecting the environmental data and updating the BIM model

According to the previous section, the light status as real-time data should be collected continuously and the BIM should be updated based on received telemetry data which constitute the DT model. According to the depicted workflow in Figure 28, the required steps will be described in detail.

#### 7.3.1. Creating the BIM model and defining the rooms

Cascina is an old project that should be renovated and it needs construction of the new rooms. new rooms should be constructed and to be joined to the existing building. Accordingly, the BIM model with the LOD 300 is constructed for the main building in Autodesk Revit based on the laser scanning survey. In addition, the new rooms are modeled based on the designer's sketch to have the whole model in one file Figure 33.



**Figure 33 - The BIM model of Cascina**

In addition, for each building that we have installed the light sensor, the mass object is modeled as the representative of the room atmosphere. Later, the properties of this mass such as color and the light status parameter will be changed based on the received telemetry data of light status.

### 7.3.2. Raspberry Pi (Hardware and Software)

Raspberry Pi is a single-board computer. That means it has all the computers components such as a processor, the graphic chip, memory, and so forth (Figure 34). Although it does not have peripheral devices such as a keyboard, mouse, or monitor, they can be connected via USB ports (Figure 35). It is also equipped with Bluetooth and a wireless chip that provides a wireless connection with other devices.



Figure 34 - Raspberry Pi board

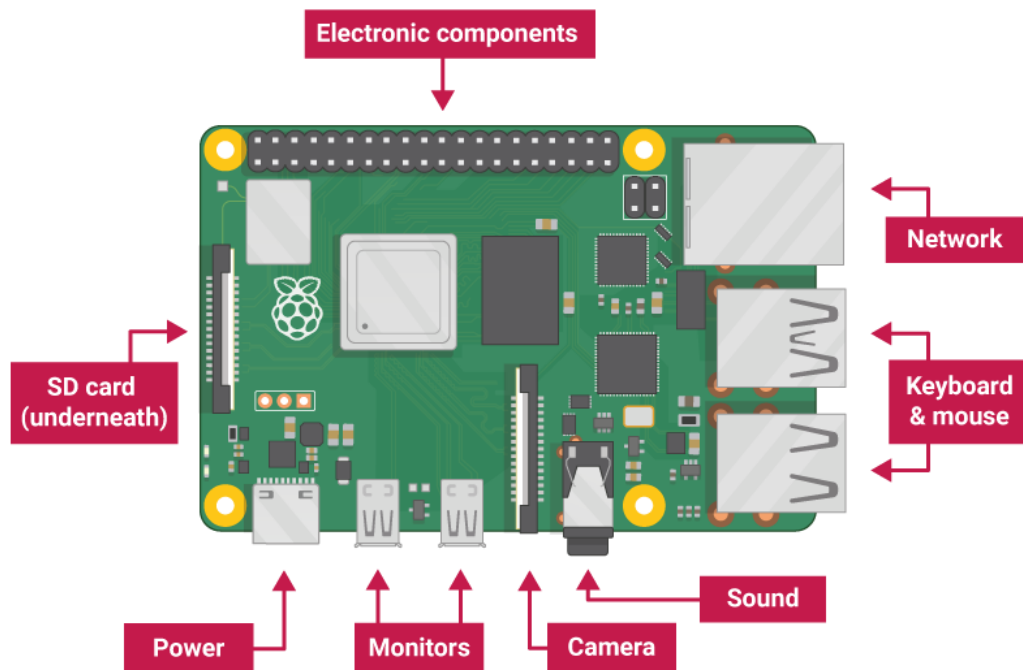


Figure 35 - Raspberry Pi ports

Raspberry Pi is a small, low-cost, and energy-efficient microcomputer which is first developed for undergraduate students. However, by the passage of time and thanks to its functionality, now it is being used in different fields and projects.

Although other single-board computers such as Arduino, Rock Pi, and so forth exist and are being used, Raspberry Pi is one of the best ones in case of functionality and speed. This small board can be used for different purposes, such as a desktop PC, a media server, an ad-blocker for your network, the brains of a robot, the monitoring system for a weather station, and so forth. One of the main Raspberry Pi features is its GPIO (General-Purpose Input/Output) port (Figure 36) which consists of 40 pins and allows you to connect electronic components such as light-emitting diodes (LEDs), buttons, sensors, and motors, as well as specialized devices known as HATs.

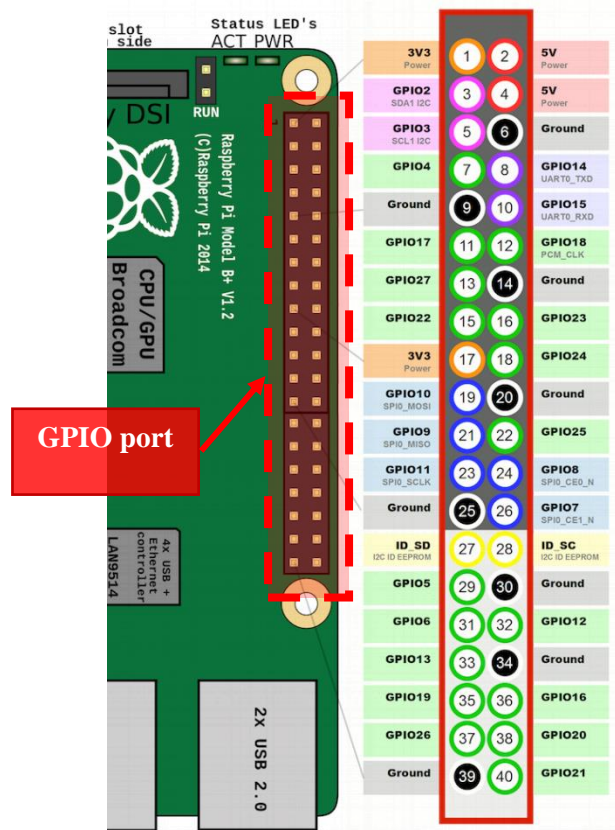


Figure 36 - Raspberry Pi GPIO port

Raspberry Pi has different models with different technical specifications. In this study, the Raspberry Pi with the following configuration is used:

- Model: Raspberry Pi model B
- CPU: Cortex A9 – Breadcom- 1.5 GHz
- RAM: DDR3 SDRAM - 8GB

As we explained before, Raspberry Pi is a complete computer, which means that we need to install Operating System (OS) for interacting with the hardware. Different OSs are released for Raspberry Pi

that are based on Linux. Linux is the OS that is mostly used on the server and World Wide Web. Among these OS, Raspbian is the most famous one which is installed and used in this study.

Although we can connect a mouse, keyboard, and monitor for working with Raspberry Pi, it is easier to use remote control software. They provide a remote connection that allows us to monitor and control the device through the internet from our laptop or PC.

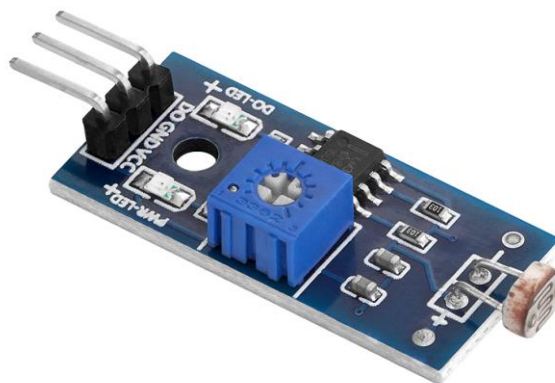
One of these remote software is VNC. VNC is a remote desktop software for cloud and offline connection. Through this app, we enable to access the Raspberry Pi desktop from our computer via the Internet and it is not necessary to work directly with that. It should be noted that other methods such as SSH can be used for remote controlling.

For using the VNC, we have to follow these processes:

1. Enabling the VNC viewer in Raspberry Pi and adjusting the resolution (Appendix 1, Figure 55)
2. Getting Raspberry Pi IP address (Appendix 1, Figure 56)
3. Installing and running the viewer on our computer and connecting to the Raspberry Pi by IP address (Appendix 1, Figure 57)
  - The application can be downloaded from the VNC website (<https://www.realvnc.com/en/connect/download/viewer/>). In the logging process, the Raspberry IP address (which was identified in the previous step) and Raspberry Pi username and password should be used

### 7.3.3. Connecting the light sensor

For sensing and collecting the environmental parameters, sensors come in hand. In this study, a Light Dependent Resistor (LDR) sensor is used to measure the indoor light intensity (Figure 37). The module sensitivity can be adjusted with an onboard potentiometer (clockwise rotation increases the sensitivity).



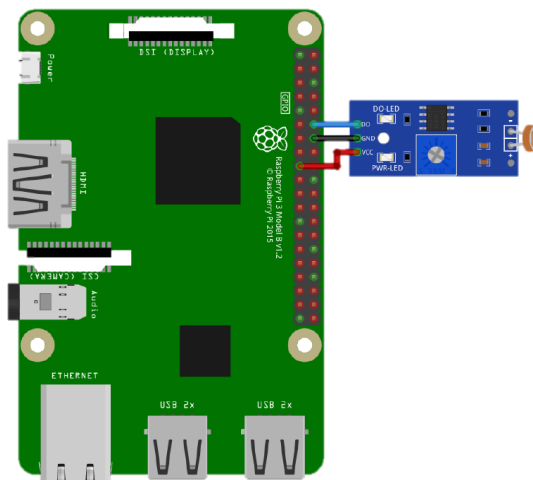
**Figure 37 - LDR sensor**

When the LDR is exposed to light, the output voltage changes due to the changes in its resistance. After amplifying the output voltage, the DO pin puts the high state and low state if it reaches and does not reach the threshold defined by the potentiometer, respectively.

This sensor has three pins:

- Digital Output (DO): It should be connected to the GPIO pins in the GPIO board
- Ground (GND): It should be connected to the GND pins in the GPIO board
- Power supply (VCC): It should be connected to the 3.3V pins in the GPIO board

It is noteworthy that we have several options for connecting these 3 pins since we have many 3.3V pins as well as some GPIO pins in the Raspberry Pi GPIO board. So, it is just important to provide the positive and negative pins and also know the GPIO number which is connected to the DO pin of the sensor. This number will be used in scripting. The sample diagram of the LDR connection is illustrated in Figure 38.



**Figure 38 - The sample diagram of LDR connection**

Based on the above information and connection to the GPIO pins, the sensor is ready to collect the light intensity.

#### **7.3.4. Reading telemetry data from the sensor and sending them to the Google Sheets via Raspberry Pi**

Reading the telemetry data from sensors and sending them to the Google Sheets is a task that should be done automatically and continuously to provide real-time data in the DT model. This process needs scripting language such as Python to facilitate the workflow.

Python which is a scripting programming language has gained popularity among developers due to its functionalities and fantastic libraries and packages. Python is an Object-Oriented Programming (OOP) that was invented by Dutch in 1991 (Kumar and Panda, 2019).

Reading the data from the sensor requires the Rpi.GPIO libraries to make a connection between our program and the GPIO board. In addition, for importing the data to the Google Sheets, we have to connect through Google Sheets API (Application Programming Interface) to the Google Sheets. API is a term in the developers' world that provides the occasion for the developers to get access to some of the software features and databases. This feature enables a developer to design their program and use

the software database. Although the developers have access to the databases through the API, they can not manipulate or use the whole database which is defined and restricted by the software companies.

Google company also provides the API that we can make a connection by scripting languages such as Python and do some tasks such as writing, reading, etc.

So, through the Python code, we can read the data from the LDR sensor and write them in the Google Sheets based on our desire frequency. The written code in Python is shown in Appendix 2, Figure 58.

### 7.3.5. Reading telemetry data from Google Sheets and storing it in a local CSV file

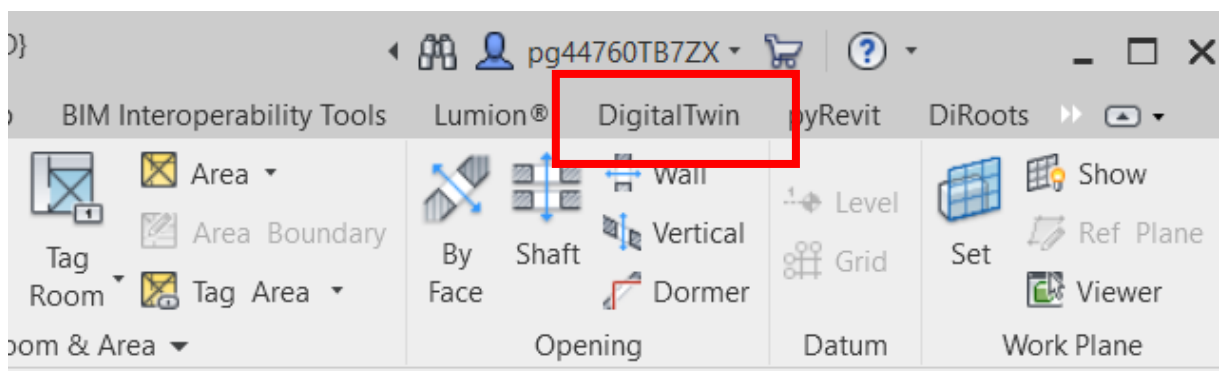
Since we have imported our data into Google Sheets, we have the access to them from the whole world. In the destination computer, the place that we have our BIM model, we can get these data through Google Sheets API and store them as a local CSV (Comma Separated Values) file. CSV is a simple text file that can be opened in a wide variety of programs such as Microsoft Excel.

To achieve this goal, we need access to Google Sheets API and CSV library. This program is written in Python and depicted in Appendix 2, Figure 59,

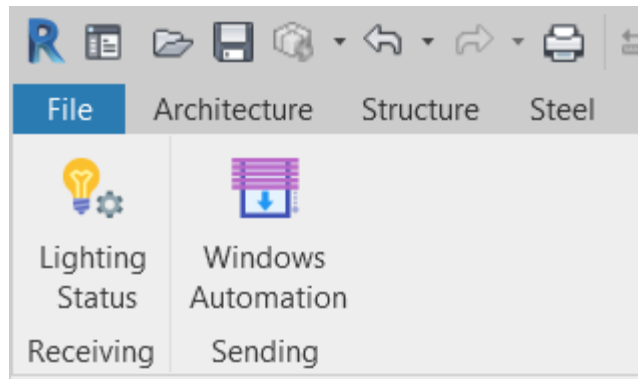
### 7.3.6. Importing the telemetry data into the Autodesk Revit and updating the BIM model

In this part we need to have the imported data of CSV file inside the Autodesk Revit, adding them to the room that the sensor is installed, and also visualizing the result. This process requires reading the data from the CSV file and working with Revit API to changing the room's parameters value and also changing the appearance of the room based on the light intensity. For sake of the users' comfort, a plugin is designed by the writer which allows users to update the BIM model inside the model by pushing the button.

The developed plugin is depicted in Figure 39. This plugin has two panels: 1) Receiving 2) Sending (Figure 40).



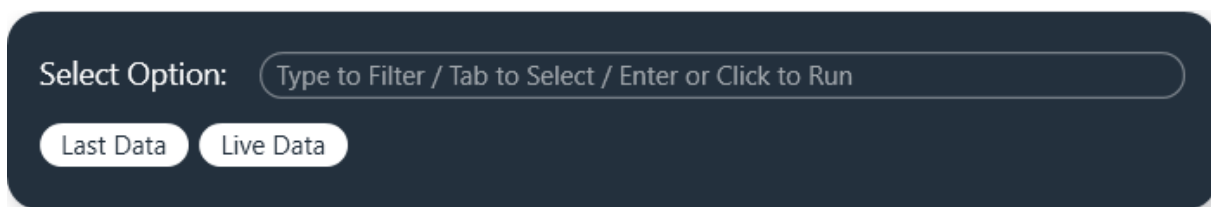
**Figure 39 – The developed plugin by the writer for updating the DT model**



**Figure 40 - The developed plugin panels**

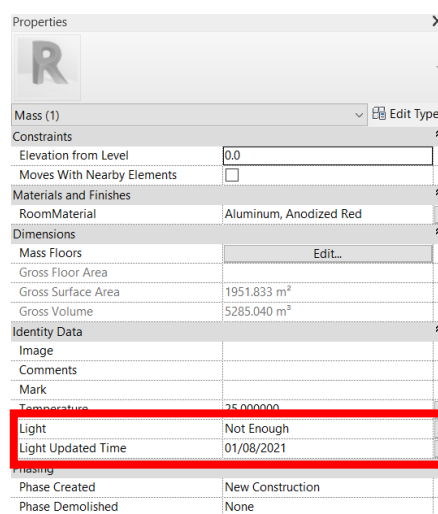
The “Receiving” panel allows the user to import the telemetry data inside the model and updating the room parameters, also, the “Sending” panel can modify the model (window family) and send required commands to the motor. The “Sending” panel will be described in detail in section 7.4.

Pushing the ‘Lighting Status’ button provides 2 options: 1) Last Data, 2) Live Data ( Figure 41).

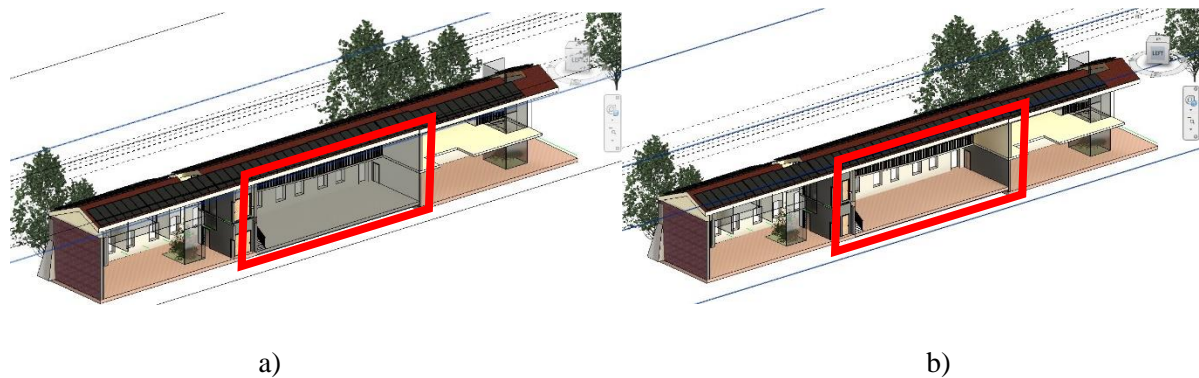


**Figure 41- Receiving options**

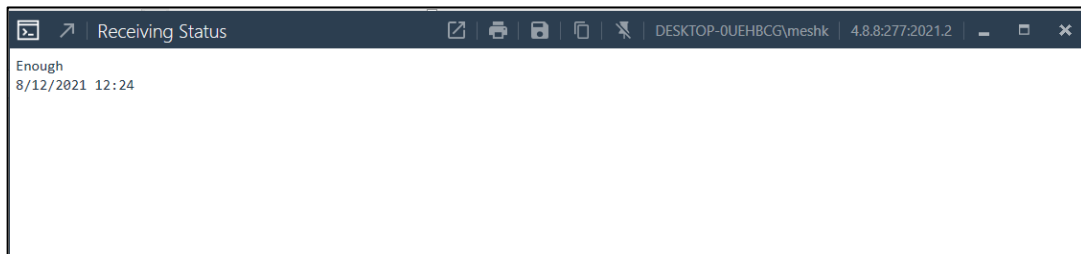
“Last Data” mode will update the values of the two parameters related to the room, “light” and “Light Updated Time” based on the real-time data (Figure 42). It also changes the room transparency in the current view. When the light is enough, the room will be transparent and when the light is not enough, the room will be presented as a solid gray mass (Figure 43). Also, these values will be shown as text in a separated window (Figure 44).



**Figure 42 - Updating the room parameters' values based on real-time data**



**Figure 43 - changing the room transparency based on the light intensity. a) light is not enough b) light is enough**



**Figure 44 - A window with light status**

In the ‘Live Data’ mode, the process of “last Data” (Updating the values, changing the room visualization, and showing the values in a separated window will be repeated for a predefined period.

Scripting the whole process comprising reading the data from the local CSV file, importing to the Autodesk Revit, updating the values, and changing the room appearance was done by using pyRevit and connection to the Revit database through Revit API.

pyRevit which is developed by Ehsan Iran-Nejad is a Rapid Application Prototyping (RAD) environment for Autodesk Revit that provides the opportunity for connecting to the Revit API using python (IronPython or CPython), C#, and VB.Net. In this study, we used IronPython for developing the plugin. The script of the “Receiving” panel is illustrated in Appendix 2, Figure 60.

#### **7.4. Getting the commands from the BIM model and sending them to the motor**

To complete the whole process of automation through the DT model, we have to be able to send commands to the real world. This section will explain the required steps for controlling the window’s louvers in a real project from the Revit in real-time status. Although the workflow is based on Figure 29, some steps will not be explained as they are similar to section 7.3.

##### **7.4.1. Creating the parametric family of the window**

As stated before, the selected movable mechanism allows louvers to move in two directions (Figure 45):

- 1) Rotary around the Z-axis
- 2) Linear along the window's plain

These movements together allow the users to select different statuses such as closing, opening, opening, semi opening, collecting the louvers on one side, etc. These features are also considered in designing the parametric model of the window (Figure 46). Parametric modeling provides an opportunity to have a digital replica of the real window and every change in the model can be reflected in a real model.

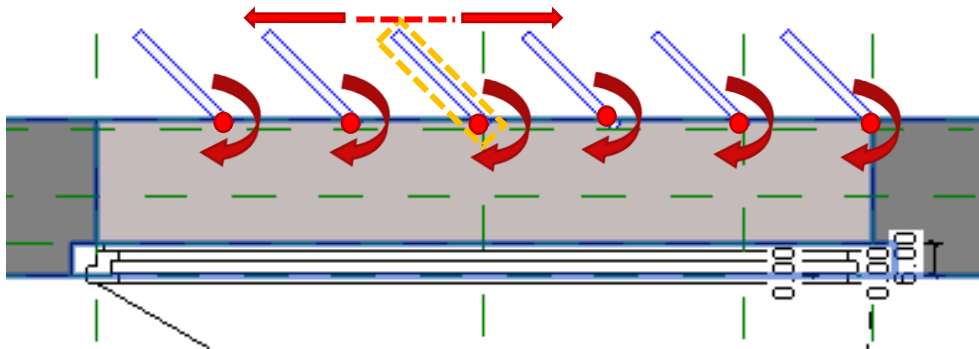


Figure 45 - Louvers' moving directions

The parametric family is designed in Autodesk Revit with the following features:

- Adjustable louvers' angles
- Adjustable louvers' distance
- Automatic definition of the number of required louvers based on the window dimensions

Although some of the above features are more than the requirements of this study, they can be used in future studies.

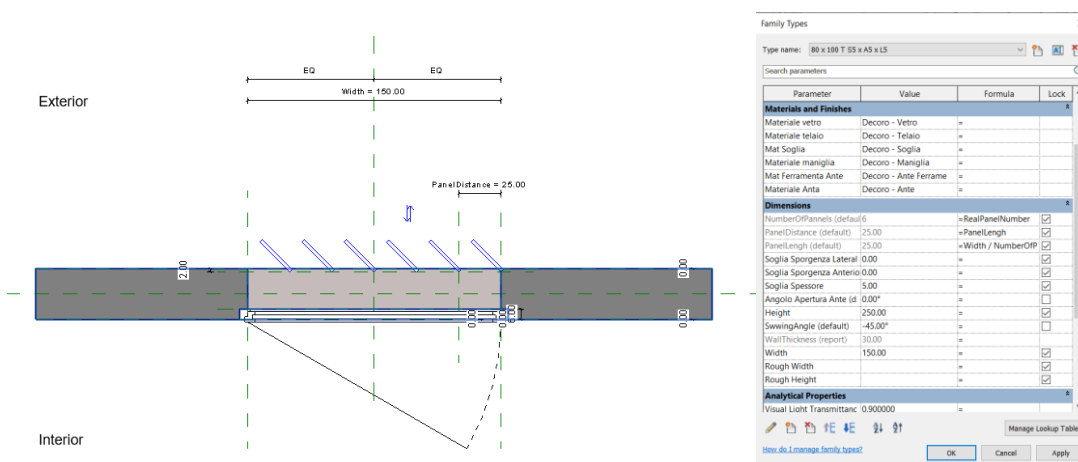


Figure 46 - Parametric model of the window

#### 7.4.2. Connecting the Motor

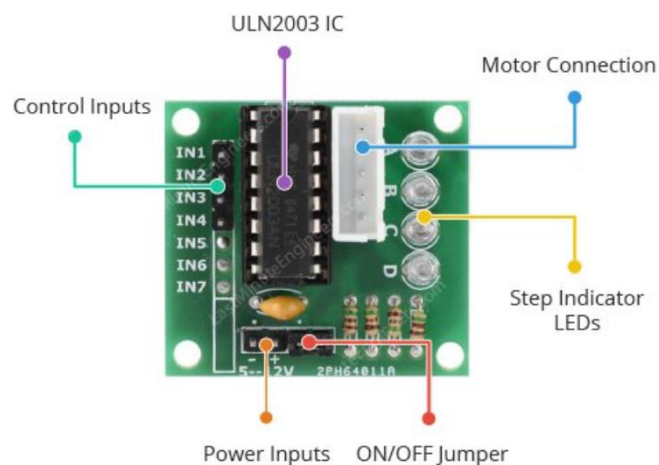
Electro motors are devices that produce mechanical movement by using electricity. These motors do not have full control over the amount of rotation. On the other hand, Stepper motors are electric motors that

divide the full rotation into equal steps. This allows us to control the position and send commands for desired moves. So, they can be used in 3D printers, CNC machines, and other devices that require precise positioning control.

In this study, the 28BYJ-48 Stepper Motor (Figure 47) is used to rotate the window louvers in the real world based on the received commands from the digital model. The motor usually comes with a ULN2003 board (Figure 48) that provides the connection of the motor to the GPIO pins and also supplies electricity.



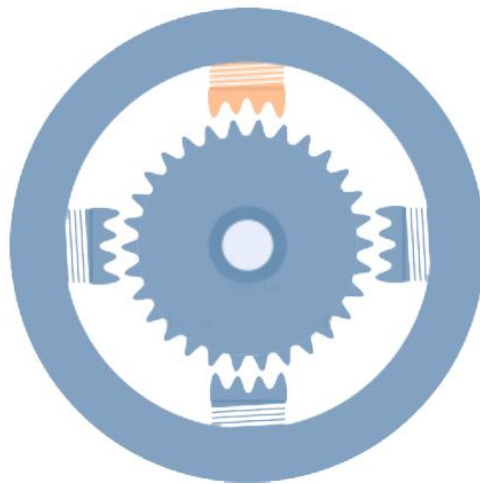
**Figure 47 - 28BYJ-48 Stepper Motor and ULN2003 board**



**Figure 48 - ULN2003 board**

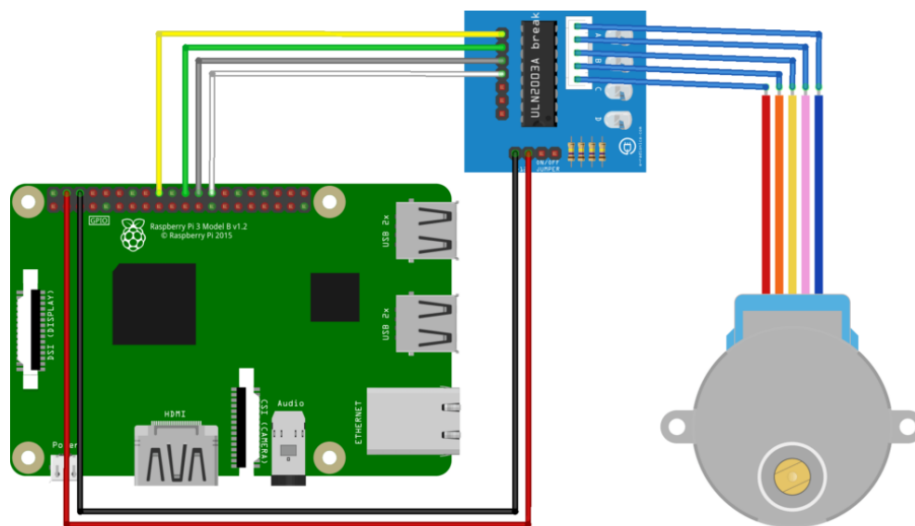
The stepper motor has a cogged wheel and four electromagnets to rotate the wheel (Figure 49). Each HIGH pulse sent, energizes the coil, attracts the nearest teeth of the cogged wheel, and drives the motor one step. Spinning direction, speed of the motor, and amount of rotation are parameters that are controlled by pulses to the coils:

- The sequence of pulses determines the spinning direction of the motor.
- The frequency of the pulses determines the speed of the motor.
- The number of pulses determines how far the motor will turn



**Figure 49 - Motor rotation mechanism**

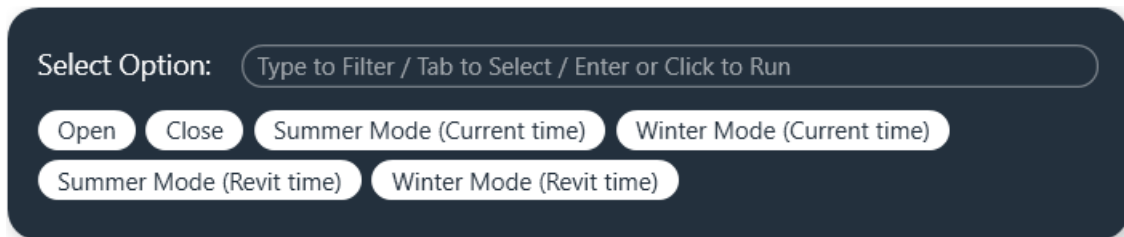
This motor needs a 5V power supply that should be noted in time of connection. Also, the four IN pins in the board should be connected to the GPIOs pins in the GPIO board of Raspberry Pi. The sample connection diagram is depicted in Figure 50.



**Figure 50 - The sample connection diagram of 28BYJ-48 and ULN2003 board to the Raspberry Pi**

#### 7.4.3. Defining scenarios for opening and closing the window's louvers

Through the automation system, we have to predict the users' desired options. Regarding that, we embedded 6 Scenarios (Figure 51):



**Figure 51 - Scenarios for moving the louvers**

- 1) “Open” mode: In this scenario regardless of the current louvers’ position, they will be rotated to open the window completely. It is clear that, if the window is opened, no moving command should be sent.
- 2) “Close” mode: In this scenario regardless of the current louvers’ position, they will be rotated to close the window completely. It is clear that, if the window is closed, no moving command should be sent.
- 3) “Summer Mode (Current time)”: In this scenario regardless of the current louvers' position, they will be rotated to the position that is perpendicular to the sun direction based on the time of the request. This scenario prepares the environment with enough light and improves users’ comfort by preventing the direct sunlight entrance and light dazzling (Figure 52).
- 4) “Winter Mode (Current time)”: In this scenario regardless of the current louvers' position, they will be rotated to the position that is parallel to the sun direction based on the time of the request. This scenario prepares the environment for the maximum sunlight entering (Figure 53).
- 5) “Summer Mode (Revit time)”: In this scenario regardless of the current louvers' position, they will be rotated to the position that is perpendicular to the sun direction based on the time that is set in “Sun Setting” in Revit. This scenario prepares the environment with enough light and improves users’ comfort by preventing the direct sunlight entrance and light dazzling (Figure 52).
- 6) “Winter Mode (Revit time)”: In this scenario regardless of the current louvers' position, they will be rotated to the position that is parallel to the sun direction based on the time that is set in “Sun Setting” in Revit. This scenario prepares the environment for the maximum sunlight entering (Figure 53).

It should be noted that in the programming part for “Summer mode” and “Winter mode”, different parameters should be considered and automated such as window direction, louvers current position, and sun position. They should be derived automatically based on the project location and project direction.

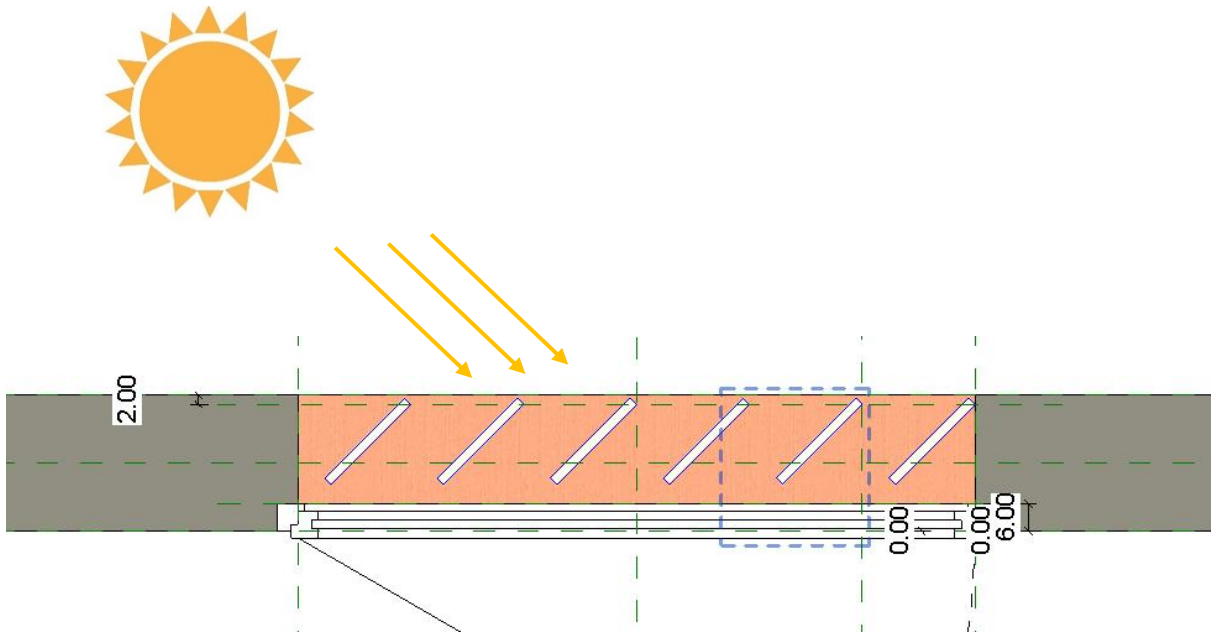


Figure 52 - Summer Mode

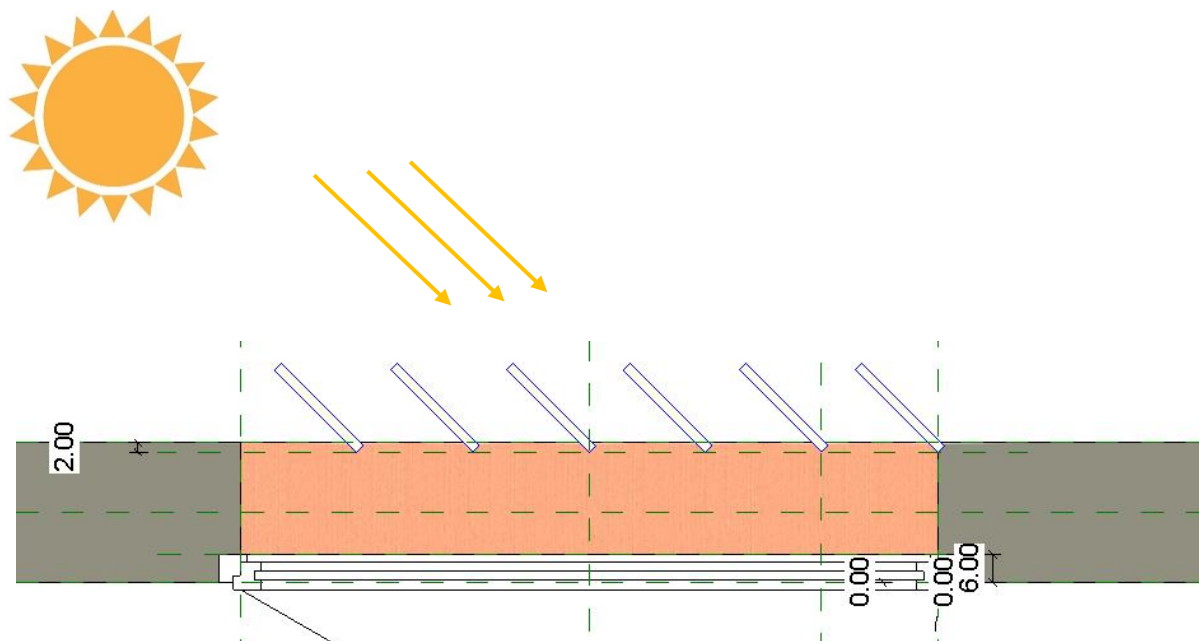


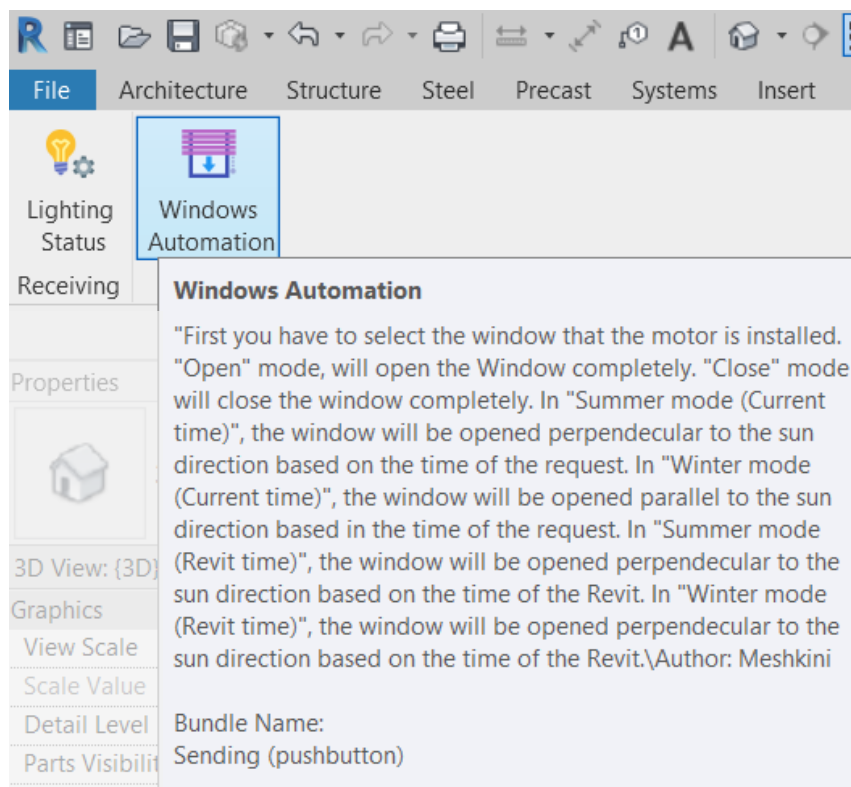
Figure 53 - Winter mode

#### 7.4.4. Modifying the model based on the scenarios, getting data from Autodesk Revit, and storing it in a local CSV file

In the previous section, we talked about different scenarios for controlling the window's louvers. These scenarios should be implemented both in the model and the window in the real world. For implementing this process, the python code is written in pyRevit. The brief features and functionality of this code are as follow:

- Easy and user-friendly menu for interacting with the user and getting the commands through the developed plugin (Figure 54) (Figure 51)
- Automatically defining the current louvers' position based on the position of the wall and relative position of the louvers
- Automatically defining the exact sun position according to the time and date of the request or the time that is set in Revit, "sun Settings"
- Updating and modifying the window model based on the selected mode (Figure 51) by the user.
- Calculating the required rotation angle and direction for the real window and writing the rotation angle as a command in the local CSV file

The written script is illustrated in Appendix 2, Figure 61.



**Figure 54 - Plug-in for updating the model and sending data to the physical model**

#### 7.4.5. Sending the telemetry data from CSV file to the Google Sheets

The required data that should be transferred is the rotation angle. This command is being stored in a local CSV file in the previous step and now it should be transferred to the Google Sheets. This step should be done continuously (Ex, every 5 seconds). This script was written in Python which can transfer data and recognize whether the value in the CSV file is the new command or not. This script is depicted in Appendix 2, Figure 62.

#### 7.4.6. Reading the data from Google Sheets and sending it to the motor

The last part for completing the process of automation is reading the commands from Google Sheets and sending them to the motor through Raspberry Pi. This step is done with Python script in Raspberry

Pi which is illustrated in Appendix 2, Figure 63. It should be noted that reading from Google Sheets should be done continuously (for example every 5 seconds) and it should not select repetitive commands if there is no new one.

## 8. CONCLUSIONS

The present study contributes to the body of knowledge by developing a DT-based process for enabling the two-direction interaction between the virtual model and the real world. Due to the importance of facility and asset management and based on the fact that the operation and maintenance phases constitute the large portion of project lifecycle cost, any improvement in this phase can affect dramatically the project objectives. Technologically, industry 4 technologies have tremendous potential for improving the industries processes and they are being developed and used in different industries. Nevertheless, The AEC/FM industry is behind other industries in adopting the technologies such as IoT, cloud computing, DT, and so forth. The main goal of this study was to fill this gap by proposing a process for using the DT features in building automation. Although some platforms are accessible in the market as a solution for adopting the DT and interacting with IoT, most of them are expensive and demands specialty for the users. The offered process and designed plugin provide the capability for the two-direction interaction in a low-cost and user-friendly environment for using daylighting.

In the current study, the importance of facility management and related concepts are explained in section 2. Section 3 introduced the BIM and its advantages that can improve the AEC/FM industry. In section 4, the industries' revolutions are described and DT, as one of the main technologies of industry 4 is introduced. Section 5 and 6 tried to introduce the main parts of the DT model, Cloud, and IoT, by explaining the concepts and their ecosystem.

In the case study, section 7, the complete process for having a DT model is proposed by integrating the 3D model and real-time telemetry data inside Autodesk Revit with the help of Python and pyRevit programming. In his study, Raspberry Pi, a single-board computer is used as a bridge for communication with sensors and the motor. The proposed platform is a low-cost method due to the use of Raspberry Pi and using the common authoring software, Autodesk Revit.

In the Cascina project, due to the importance of daylighting, the installed lighting sensor can transmit the daylighting intensity through the Raspberry Pi to the Google Sheets and integrate with the 3D model in Autodesk Revit with the help of a designed plugin by Python and pyRevit. In addition, based on the defined scenarios, the window's louvers can rotate in the real world and virtual model based on the command issued from Autodesk Revit or smartphone within the automation system for optimizing the daylighting as well as improving the occupants' satisfaction. The key contribution of the proposed process is its low price and user-friendly plugin.

The proposed process is capable of adding other sensors such as humidity, temperature, Co2 detection, and so forth. All of these telemetry data can be collected, monitored, visualized, and integrated with the 3D model through this process.

Opportunities are available to improve the proposed process in future works. First, more sensors can be installed in the project for having more complete information about IEQ which helps the users for decision making. Second, other storing software or databases can be used instead of Google Sheets to have a more structuralized data repository. In addition, designing an online 3D viewer instead of using the Autodesk Revit can dramatically improve its functionality.

This page is intentionally left blank

## REFERENCES

- Aaron, C. and Charles, E. (2019) 'Need for Interoperability to Enable Seamless Information Exchanges in Smart and Sustainable Urban Systems', *Journal of Computing in Civil Engineering*, 33(3), p. 4019008. doi: 10.1061/(ASCE)CP.1943-5487.0000824.
- Aceto, G., Persico, V. and Pescapé, A. (2020) 'Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0', *Journal of Industrial Information Integration*, 18, p. 100129. doi: <https://doi.org/10.1016/j.jii.2020.100129>.
- Aheleroff, S. *et al.* (2020) 'IoT-enabled smart appliances under industry 4.0: A case study', *Advanced Engineering Informatics*, 43, p. 101043. doi: 10.1016/j.aei.2020.101043.
- Aheleroff, S. *et al.* (2021) 'Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model', *Advanced Engineering Informatics*, 47, p. 101225. doi: <https://doi.org/10.1016/j.aei.2020.101225>.
- Alaloul, W. S. *et al.* (2018) 'Industry Revolution IR 4.0: Future Opportunities and Challenges in Construction Industry', *MATEC Web Conf.*, 203. Available at: <https://doi.org/10.1051/mateconf/201820302010>.
- Alaloul, W. S. *et al.* (2020) 'Industrial Revolution 4.0 in the construction industry: Challenges and opportunities for stakeholders', *Ain Shams Engineering Journal*, 11(1), pp. 225–230. doi: <https://doi.org/10.1016/j.asej.2019.08.010>.
- Alvarez-Romero, S. (2014) 'Use of Building Information Modeling technology in the integration of the handover process and facilities management', in.
- Antony Franklin, A. and Tambe, S. D. (2020) 'Multi-access edge computing in cellular networks', *CSI Transactions on ICT*, 8(1), pp. 85–92. doi: 10.1007/s40012-020-00276-6.
- Apanaviciene, R., Vanagas, A. and Fokaides, P. A. (2020) 'Smart Building Integration into a Smart City (SBISC): Development of a New Evaluation Framework', *Energies*. doi: 10.3390/en13092190.
- Aslanpour, M. S., Gill, S. S. and Toosi, A. N. (2020) 'Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research', *Internet of Things*, 12, p. 100273. doi: <https://doi.org/10.1016/j.iot.2020.100273>.
- Atlam, H. F. *et al.* (2020) 'Internet of Things Forensics: A Review', *Internet of Things*, 11, p. 100220. doi: 10.1016/j.iot.2020.100220.
- Atlam, H. F., Walters, R. and Wills, G. (2018) 'Internet of Things: state-of-the-art, challenges, applications, and open issues', in.
- Aziz, N. D., Nawawi, A. H. and Ariff, N. R. M. (2016) 'Building Information Modelling (BIM) in Facilities Management: Opportunities to be Considered by Facility Managers', *Procedia - Social and Behavioral Sciences*, 234, pp. 353–362. doi: 10.1016/j.sbspro.2016.10.252.
- Bansal, S. and Kumar, D. (2020) 'IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication', *International Journal of Wireless Information Networks*, 27(3), pp. 340–364. doi: 10.1007/s10776-020-00483-7.
- Bittencourt, L. *et al.* (2018) 'The Internet of Things, Fog and Cloud continuum: Integration and challenges', *Internet of Things*, 3–4, pp. 134–155. doi:

<https://doi.org/10.1016/j.iot.2018.09.005>.

- BSI Standards Limited (2013) *PAS 1192-2: Specification for information management for the capital/delivery phase of construction projects using building information modelling*.
- Caiza, G. *et al.* (2020) ‘Fog computing at industrial level, architecture, latency, energy, and security: A review’, *Heliyon*, 6(4), p. e03706. doi: <https://doi.org/10.1016/j.heliyon.2020.e03706>.
- Carvalho, N. *et al.* (2018) ‘Manufacturing in the fourth industrial revolution: A positive prospect in Sustainable Manufacturing’, *Procedia Manufacturing*, 21, pp. 671–678. doi: [10.1016/j.promfg.2018.02.170](https://doi.org/10.1016/j.promfg.2018.02.170).
- Cato, W. and Mobley, K. (2002) *Computer-Managed Maintenance Systems*. 2nd edn. doi: <https://doi.org/10.1016/B978-0-7506-7473-7.X5000-1>.
- Chalmers, D. and Sloman, M. (1999) ‘A survey of quality of service in mobile computing environments’, *IEEE Communications Surveys*, 2(2), pp. 2–10. doi: [10.1109/COMST.1999.5340514](https://doi.org/10.1109/COMST.1999.5340514).
- Cohen, T., Baretich, M. F. and Gentles, W. M. (2020) ‘Chapter 34 - Computerized maintenance management systems’, in Iadanza, E. B. T.-C. E. H. (Second E. (ed.). Academic Press, pp. 208–218. doi: <https://doi.org/10.1016/B978-0-12-813467-2.00034-1>.
- Coskun, V., Ozdenizci, B. and Ok, K. (2013) ‘A Survey on Near Field Communication (NFC) Technology’, *Wireless Personal Communications*, 71(3), pp. 2259–2294. doi: [10.1007/s11277-012-0935-5](https://doi.org/10.1007/s11277-012-0935-5).
- Costin, A. (2016) *A NEW METHODOLOGY FOR INTEROPERABILITY OF HETEROGENEOUS BRIDGE INFORMATION MODELS*. Georgia Institute of Technology.
- Al Dakheel, J. *et al.* (2020) ‘Smart buildings features and key performance indicators: A review’, *Sustainable Cities and Society*, 61, p. 102328. doi: <https://doi.org/10.1016/j.scs.2020.102328>.
- Dall’O’, G., Zichi, A. and Torri, M. (2020) ‘Green BIM and CIM: Sustainable Planning Using Building Information Modelling BT - Green Planning for Cities and Communities: Novel Incisive Approaches to Sustainability’, in Dall’O’, G. (ed.). Cham: Springer International Publishing, pp. 383–409. doi: [10.1007/978-3-030-41072-8\\_17](https://doi.org/10.1007/978-3-030-41072-8_17).
- Dastjerdi, A. V and Buyya, R. (2016) ‘Fog Computing: Helping the Internet of Things Realize Its Potential’, *Computer*, 49(8), pp. 112–116. doi: [10.1109/MC.2016.245](https://doi.org/10.1109/MC.2016.245).
- Dillon, T., Wu, C. and Chang, E. (2010) ‘Cloud Computing: Issues and Challenges’, in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 27–33. doi: [10.1109/AINA.2010.187](https://doi.org/10.1109/AINA.2010.187).
- Dolui, K. and Datta, S. K. (2017) ‘Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing’, in *2017 Global Internet of Things Summit (GIoTS)*, pp. 1–6. doi: [10.1109/GIOTS.2017.8016213](https://doi.org/10.1109/GIOTS.2017.8016213).
- Donno, M. De, Tange, K. and Dragoni, N. (2019) ‘Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog’, *IEEE Access*, 7, pp. 150936–150948. doi: [10.1109/ACCESS.2019.2947652](https://doi.org/10.1109/ACCESS.2019.2947652).
- Dujovne, D. *et al.* (2014) ‘6TiSCH: deterministic IP-enabled industrial internet (of things)’, *IEEE Communications Magazine*, 52(12), pp. 36–41. doi: [10.1109/MCOM.2014.6979984](https://doi.org/10.1109/MCOM.2014.6979984).

- Dupláková, D. *et al.* (2019) 'Ergonomic rationalization of lighting in the working environment. Part I: Proposal of rationalization algorithm for lighting redesign', *International Journal of Industrial Ergonomics*, 71, pp. 92–102. doi: <https://doi.org/10.1016/j.ergon.2019.02.012>.
- E., E. *et al.* (2021) 'Formalization of Occupant Feedback and Integration with BIM in Office Buildings', *Journal of Computing in Civil Engineering*, 35(1), p. 4020055. doi: 10.1061/(ASCE)CP.1943-5487.0000940.
- Eastman, C. *et al.* (2018) *BIM Handbook: a Guide to Building Information Modeling for Owners, Managers, Architects, Engineers, Contractors, and Fabricators*. Newark: John Wiley & Sons, Incorporated.
- Gao, X. and Pishdad-Bozorgi, P. (2020) 'A framework of developing machine learning models for facility life-cycle cost analysis', *Building Research & Information*, 48(5), pp. 501–525. doi: 10.1080/09613218.2019.1691488.
- Geng, Y. *et al.* (2017) 'The impact of thermal environment on occupant IEQ perception and productivity', *Building and Environment*, 121, pp. 158–167. doi: 10.1016/j.buildenv.2017.05.022.
- Ghani, N. A. M. *et al.* (2020) 'Artificial Neural Network Analysis on BMS dataset using Scale Conjugate and Gradient Descent Optimization Techniques', in *2020 IEEE Conference on Big Data and Analytics (ICBDA)*, pp. 51–56. doi: 10.1109/ICBDA50157.2020.9289759.
- Ghobakhloo, M. *et al.* (2021) 'Industry 4.0 ten years on: A bibliometric and systematic review of concepts, sustainability value drivers, and success determinants', *Journal of Cleaner Production*, 302, p. 127052. doi: <https://doi.org/10.1016/j.jclepro.2021.127052>.
- Guenther, R. and Radebaugh, J. (2004) *Understanding Metadata*. Available at: [www.niso.org](http://www.niso.org).
- Hashem, I. A. T. *et al.* (2015) 'The rise of "big data" on cloud computing: Review and open research issues', *Information Systems*, 47, pp. 98–115. doi: <https://doi.org/10.1016/j.is.2014.07.006>.
- He, Z. Y. and Liang, Y. (2015) 'Study on the DDS Network Information Security Technology', *Applied Mechanics and Materials*, 738–739, pp. 1213–1216. doi: 10.4028/www.scientific.net/AMM.738-739.1213.
- Herr, C. M. and Fischer, T. (2019) 'BIM adoption across the Chinese AEC industries: An extended BIM adoption model', *Journal of Computational Design and Engineering*, 6(2), pp. 173–178. doi: 10.1016/j.jcde.2018.06.001.
- Hexu, L., Osama, A. and William, L. (2021) 'BIM-Based Smart Facility Management: A Review of Present Research Status, Challenges, and Future Needs', *Construction Research Congress 2020*. (Proceedings), pp. 1087–1095. doi: 10.1061/9780784482865.115.
- Hilal, M., Maqsood, T. and Abdekhodae, A. (2019) 'A scientometric analysis of BIM studies in facilities management', *International Journal of Building Pathology and Adaptation*, 37(2), pp. 122–139. doi: 10.1108/IJBPA-04-2018-0035.
- Hofer-Schmitz, K. and Stojanović, B. (2020) 'Towards formal verification of IoT protocols: A Review', *Computer Networks*, 174, p. 107233. doi: <https://doi.org/10.1016/j.comnet.2020.107233>.
- Hossain, M. F. (2019) 'Chapter Seven - Best Management Practices', in Hossain, M. F. B. T.-S. D. and B. (ed.). Butterworth-Heinemann, pp. 419–431. doi: <https://doi.org/10.1016/B978-0-12-816722-9.00007-0>.

- Hyunji, S. *et al.* (2018) 'Facility Management Process of an Office Building', *Journal of Infrastructure Systems*, 24(3), p. 4018017. doi: 10.1061/(ASCE)IS.1943-555X.0000436.
- ifma (2020) *IFMA*. Available at: <https://ftmaintenance.com/maintenance-management/what-is-facility-management/>.
- International Organization for Standardization (ISO) 41011 (2017) *No Title*.
- Jincy, V. J. and Sundararajan, S. (2015) 'Classification Mechanism for IoT Devices towards Creating a Security Framework BT - Intelligent Distributed Computing', in Buyya, R. and Thampi, S. M. (eds). Cham: Springer International Publishing, pp. 265–277.
- Kallio, J. *et al.* (2020) 'Assessment of perceived indoor environmental quality, stress and productivity based on environmental sensor data and personality categorization', *Building and Environment*, 175, p. 106787. doi: 10.1016/j.buildenv.2020.106787.
- Karim, F. *et al.* (2018) 'Taxonomy for BIM and Asset Management Semantic Interoperability', *Journal of Management in Engineering*, 34(4), p. 4018012. doi: 10.1061/(ASCE)ME.1943-5479.0000610.
- Karlapudi, J. *et al.* (2020) 'Enhancement of BIM Data Representation in Product-Process Modelling for Building Renovation BT - Product Lifecycle Management Enabling Smart X', in Nyffenegger, F. *et al.* (eds). Cham: Springer International Publishing, pp. 738–752.
- Kavis, M. J. (2014) *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)*. Wiley.
- Kim, J. and de Dear, R. (2012) 'Nonlinear relationships between individual IEQ factors and overall workspace satisfaction', *Building and Environment*, 49(1), pp. 33–40. doi: 10.1016/j.buildenv.2011.09.022.
- Koch, C. *et al.* (2014) 'Natural markers for augmented reality-based indoor navigation and facility maintenance', *Automation in Construction*, 48, pp. 18–30. doi: 10.1016/j.autcon.2014.08.009.
- Korsavi, S. S., Montazami, A. and Mumovic, D. (2020) 'The impact of indoor environment quality (IEQ) on school children's overall comfort in the UK; a regression approach', *Building and Environment*, 185, p. 107309. doi: 10.1016/j.buildenv.2020.107309.
- Kumar, A. and Panda, S. P. (2019) 'A Survey: How Python Pitches in IT-World', in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 248–251. doi: 10.1109/COMITCon.2019.8862251.
- Lee, D. *et al.* (2021) 'Integrated digital twin and blockchain framework to support accountable information sharing in construction projects', *Automation in Construction*, 127, p. 103688. doi: <https://doi.org/10.1016/j.autcon.2021.103688>.
- Li, Y. and Liu, A. (2013) *Analysis of the Challenges and Solutions of Building a Smart City*. doi: 10.1061/9780784413135.149.
- Liu, L., Wang, H. and Zhang, Y. (2020) 'Secure IoT Data Outsourcing With Aggregate Statistics and Fine-Grained Access Control', *IEEE Access*, 8, pp. 95057–95067. doi: 10.1109/ACCESS.2019.2961413.
- Lopes, I. *et al.* (2016) 'Requirements Specification of a Computerized Maintenance Management System – A Case Study', *Procedia CIRP*, 52, pp. 268–273. doi: <https://doi.org/10.1016/j.procir.2016.07.047>.

- Lu, Q. *et al.* (2020) ‘Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance’, *Automation in Construction*, 118, p. 103277. doi: <https://doi.org/10.1016/j.autcon.2020.103277>.
- Maqsoom, A. *et al.* (2020) ‘Exploring managerial perspectives of using building management system through tam: An empirical study of commercial sector of pakistan’, *Periodica Polytechnica Civil Engineering*, 64(3), pp. 690–701. doi: 10.3311/PPci.15323.
- Michalec, A. (Ola), Hayes, E. and Longhurst, J. (2019) ‘Building smart cities, the just way. A critical review of “smart” and “just” initiatives in Bristol, UK’, *Sustainable Cities and Society*, 47, p. 101510. doi: <https://doi.org/10.1016/j.scs.2019.101510>.
- Munyensanga, P. *et al.* (2018) ‘Information management to improve the effectiveness of preventive maintenance activities with computerized maintenance management system at the intake system of circulating water pump’, *Procedia CIRP*, 78, pp. 289–294. doi: <https://doi.org/10.1016/j.procir.2018.09.044>.
- Mustafa, C. and Zeebaree, S. (2021) ‘Sufficient Comparison Among Cloud Computing Services: IaaS, PaaS, and SaaS: A Review’, 5, pp. 17–30. doi: 10.5281/zenodo.4481415.
- Naren, J., Sowmya, S. K. and Deepika, P. (2014) ‘Layers of Cloud – IaaS, PaaS and SaaS: A Survey’, *International Journal of Computer Science and Information Technology*, Vol. 5 (3), pp. 4477–4480.
- NBIMS, N. B. I. M. S. (2010) *National Building Information Modeling Standard*.
- Oliveira, L. *et al.* (2019) ‘MAC Layer Protocols for Internet of Things: A Survey’, *Future Internet*. doi: 10.3390/fi11010016.
- Onile, A. E. *et al.* (2021) ‘Uses of the digital twins concept for energy services, intelligent recommendation systems, and demand side management: A review’, *Energy Reports*, 7, pp. 997–1015. doi: <https://doi.org/10.1016/j.egyr.2021.01.090>.
- Pan, Y. and Zhang, L. (2021) ‘A BIM-data mining integrated digital twin framework for advanced project management’, *Automation in Construction*, 124, p. 103564. doi: <https://doi.org/10.1016/j.autcon.2021.103564>.
- Pardis, P.-B. (2017) ‘Future Smart Facilities: State-of-the-Art BIM-Enabled Facility Management’, *Journal of Construction Engineering and Management*, 143(9), p. 2517006. doi: 10.1061/(ASCE)CO.1943-7862.0001376.
- Parkinson, T., Parkinson, A. and de Dear, R. (2019) ‘Continuous IEQ monitoring system: Context and development’, *Building and Environment*, 149, pp. 15–25. doi: 10.1016/j.buildenv.2018.12.010.
- PlanGrid (2018) *2018 industry report: construction disconnected*.
- Ramya, C. M., Shanmugaraj, M. and Prabakaran, R. (2011) ‘Study on ZigBee technology’, in *2011 3rd International Conference on Electronics Computer Technology*, pp. 297–301. doi: 10.1109/ICECTECH.2011.5942102.
- Rasheed, E. O., Khoshbakht, M. and Baird, G. (2021) ‘Time spent in the office and workers’ productivity, comfort and health: A perception study’, *Building and Environment*, 195, p. 107747. doi: 10.1016/j.buildenv.2021.107747.
- Ray, A. K. and Bagwari, A. (2017) ‘Study of smart home communication protocol’s and security & privacy aspects’, in *2017 7th International Conference on Communication Systems and Network*

*Technologies (CSNT)*, pp. 240–245. doi: 10.1109/CSNT.2017.8418545.

- Reza, H. *et al.* (2015) ‘Barriers to Adoption of Building Information Modeling (BIM) in Iran: Preliminary Results’, in:
- Roche, S. *et al.* (2012) ‘Are “Smart Cities” Smart Enough?’, *Spatially Enabling Government, Industry and Citizens: Research Development and Perspectives*, pp. 215–236.
- Sadick, A. M., Kpamma, Z. E. and Agyefi-Mensah, S. (2020) ‘Impact of indoor environmental quality on job satisfaction and self-reported productivity of university employees in a tropical African climate’, *Building and Environment*, 181, p. 107102. doi: 10.1016/j.buildenv.2020.107102.
- Saka, A. B. and Chan, D. W. M. (2020) ‘Profound barriers to building information modelling (BIM) adoption in construction small and medium-sized enterprises (SMEs)’, *Construction Innovation*, 20(2), pp. 261–284. doi: 10.1108/CI-09-2019-0087.
- Saratu, T., Chimay, A. and Somayeh, A. (2021) ‘BIM Implementation in Facilities Management: An Analysis of Implementation Processes’, *Construction Research Congress 2018. (Proceedings)*, pp. 725–735. doi: doi:10.1061/9780784481264.071.
- Schleich, B. *et al.* (2017) ‘Shaping the digital twin for design and production engineering’, *CIRP Annals - Manufacturing Technology*, 66, pp. 141–144. doi: 10.1016/j.cirp.2017.04.040.
- Semeraro, C. *et al.* (2021) ‘Digital twin paradigm: A systematic literature review’, *Computers in Industry*, 130, p. 103469. doi: https://doi.org/10.1016/j.compind.2021.103469.
- Shallal, Q. and Bokhari, M. (2019) ‘A Survey on Cloud Computing’.
- Shi, W. and Dustdar, S. (2016) ‘The Promise of Edge Computing’, *Computer*, 49(5), pp. 78–81. doi: 10.1109/MC.2016.145.
- Suresh, P. *et al.* (2014) ‘A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment’, in *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, pp. 1–8. doi: 10.1109/ICSEMR.2014.7043637.
- Tao, F. *et al.* (2019) ‘Digital Twin in Industry: State-of-the-Art’, *IEEE Transactions on Industrial Informatics*, 15(4), pp. 2405–2415. doi: 10.1109/TII.2018.2873186.
- Ting-Kwei, W. and Yanmei, P. (2019) ‘Development of BIM-AR-Based Facility Risk Assessment and Maintenance System’, *Journal of Performance of Constructed Facilities*, 33(6), p. 4019068. doi: 10.1061/(ASCE)CF.1943-5509.0001339.
- Tinmaz, H. (2020) ‘History of Industrial Revolutions: From Homo Sapiens Hunters to Bitcoin Hunters BT - Blockchain Technology for Industry 4.0: Secure, Decentralized, Distributed and Trusted Industry Environment’, in Rosa Righi, R. da, Alberti, A. M., and Singh, M. (eds). Singapore: Springer Singapore, pp. 1–26. doi: 10.1007/978-981-15-1137-0\_1.
- Tournier, J. *et al.* (2020) ‘A survey of IoT protocols and their security issues through the lens of a generic IoT stack’, *Internet of Things*, p. 100264. doi: 10.1016/j.iot.2020.100264.
- Turk, Ž. (2016) ‘Ten questions concerning building information modelling’, *Building and Environment*, 107, pp. 274–284. doi: 10.1016/j.buildenv.2016.08.001.
- Ullah, K., Lill, I. and Witt, E. (2019) ‘An Overview of BIM Adoption in the Construction Industry: Benefits and Barriers’, in Lill, I. and Witt, E. (eds) *10th Nordic Conference on Construction Economics and Organization*. Emerald Publishing Limited (Emerald Reach Proceedings

- Series), pp. 297–303. doi: 10.1108/S2516-285320190000002052.
- United nations (2019) *World Urbanization Prospects*. New York, NY, USA. Available at: <https://population.un.org/wup/Publications/>.
- Vaidya, S., Ambad, P. and Bhosle, S. (2018) ‘Industry 4.0 - A Glimpse’, in *Procedia Manufacturing*. Elsevier B.V., pp. 233–238. doi: 10.1016/j.promfg.2018.02.034.
- Volk, R., Stengel, J. and Schultmann, F. (2014) ‘Building Information Modeling (BIM) for existing buildings — Literature review and future needs’, *Automation in Construction*, 38, pp. 109–127. doi: <https://doi.org/10.1016/j.autcon.2013.10.023>.
- Wienker, M., Henderson, K. and Volkerts, J. (2016) ‘The Computerized Maintenance Management System an Essential Tool for World Class Maintenance’, *Procedia Engineering*, 138, pp. 413–420. doi: <https://doi.org/10.1016/j.proeng.2016.02.100>.
- Wikipedia (2021) *operability definition*. Available at: <https://en.wikipedia.org/wiki/Operability>.
- Xinghua, G. *et al.* (2021) ‘Internet of Things Enabled Data Acquisition Framework for Smart Building Applications’, *Journal of Construction Engineering and Management*, 147(2), p. 4020169. doi: 10.1061/(ASCE)CO.1943-7862.0001983.
- Yang, C., Xinyi, S. and Xuan, J. (2021) ‘An Agent-Based Framework for Occupant-Oriented Intelligent Facility Management Scheduling’, *Computing in Civil and Building Engineering (2014)*. (Proceedings), pp. 1828–1835. doi: doi:10.1061/9780784413616.227.
- Yang, S. *et al.* (2020) ‘Experimental study of model predictive control for an air-conditioning system with dedicated outdoor air system’, *Applied Energy*, 257, p. 113920. doi: <https://doi.org/10.1016/j.apenergy.2019.113920>.
- Yassein, M. B., Shatnawi, M. Q. and Al-zoubi, D. (2016) ‘Application layer protocols for the Internet of Things: A survey’, in *2016 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–4. doi: 10.1109/ICEMIS.2016.7745303.
- Yi, S. *et al.* (2015) ‘Fog Computing: Platform and Applications’, in *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, pp. 73–78. doi: 10.1109/HotWeb.2015.22.
- Yousefpour, A. *et al.* (2019) ‘All one needs to know about fog computing and related edge computing paradigms: A complete survey’, *Journal of Systems Architecture*, 98, pp. 289–330. doi: <https://doi.org/10.1016/j.sysarc.2019.02.009>.
- Yugha, R. and Chithra, S. (2020) ‘A survey on technologies and security protocols: Reference for future generation IoT’, *Journal of Network and Computer Applications*, 169, p. 102763. doi: <https://doi.org/10.1016/j.jnca.2020.102763>.
- Z., C., L., A. and K., M. (2021) ‘Business Models for Decentralised Facility Management Supported by Radio Frequency Identification Technology’, *Computing in Civil Engineering (2011)*. (Proceedings), pp. 323–330. doi: doi:10.1061/41182(416)40.
- Zembrane, H., Baddi, Y. and Hasbi, A. (2018) ‘Comparison between IOT Protocols: ZigBee and WiFi Using the OPNET Simulator’, in *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*. New York, NY, USA: Association for Computing Machinery (SITA’18). doi: 10.1145/3289402.3289522.
- Zeng, M. (2019) ‘Interoperability’, (Knowledge Organization 46, 2), pp. 122–146. Available at: <https://www.isko.org/index.php>.

Zhan, S., Chong, A. and Lasternas, B. (2021) 'Automated recognition and mapping of building management system (BMS) data points for building energy modeling (BEM)', *Building Simulation*, 14(1), pp. 43–52. doi: 10.1007/s12273-020-0612-7.

## LIST OF ACRONYMS AND ABBREVIATIONS

AMQP	Advanced Message Queuing Protocol
AHP	Analytic Hierarchy Process
API	Application Programming Interface
AEC	Architecture, Engineering, and Construction
AEC/FM	Architecture, Engineering, and Construction, Facility Management
AI	Artificial Intelligence
ANN	Artificial Neural Network
BOCPD	Bayesian Online Changepoint Detection
BEP	BIM Execution Plan
BAS	Building Automation System
BEM	Building Energy Modeling
BIM	Building Information Modelling
BMS	Building Management System
CARP	Channel-Aware Routing Protocol
CORPL	Cognitive and Opportunistic RPL
CSV	Comma Separated Values
CDE	Common Data Environment
CAFM	Computer-Aided Facility Management
CMMS	Computerized Maintenance Management System
CoAP	Constrained Application Protocol
CRM	Customer Relationship Management
CPS	Cyber-Physical Systems
DDS	Data Distribution Service
DO	Digital Output
DT	Digital Twin
EC	Edge Computing
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
FM	Facility Management
FaaS	Function as a Service
GPIO	General-Purpose Input/Output
GPS	Global Positioning System
GND	Ground
H2H	Human-to-Human
H2M	Human-to-Machine
IEQ	Indoor Environmental Quality
IFC	Industrial Foundation Class
ICT	Information and Communications Technology
IaaS	Infrastructure as a Service
ISO	International Standards Organization
IETF	Internet Engineering Task Force
IoT	Internet of Things
Ipv6	Internet Protocol version 6

6LowPAN	IPv6 low power wireless personal area networks
LOD	Level of Definition
LOD	Level of detail
LOD	Level of Development
LOD	Level of Development of Objects
LOIN	Level Of Information Need
LDR	Light Dependent Resistor
LEDs	light-emitting diodes
LAN	Local Area Networks
LoRaWAN	Long-Range Wireless Area Network
LPWAN	Low-Power Wide-Area Network
M2M	Machin-to-Machin
MAC	Media Address Control
MQTT	Message Queue Telemetry Transport
MACC	Mobile ad hoc Cloud Computing
MCC	Mobile Cloud Computing
MC	Mobile Computing
MEC	Multi-access Edge Computing
MCDM	Multiple-Criteria Decision-Making
NFC	Near Field Communication
NZEB	Nearly Zero Energy Building
OOP	Object-Oriented Programming
OS	Operating Service
O&M	Operation and Maintenance
PaaS	Platform as a Service
QoS	Quality of Service
RFID	Radio Frequent Identification
RAD	Rapid Application Prototyping
RPL	Routine Protocol
SC	Smart City
SaaS	Software as a Service
UAV	Unmanned Aerial Vehicle
WAN	Wide Area Networks
WPAN	Wireless Personal Area Network
Wi-Fi	Wireless-Fidelity

This page is intentionally left blank

# APPENDICES

## APPENDIX 1: VNC VIEWER

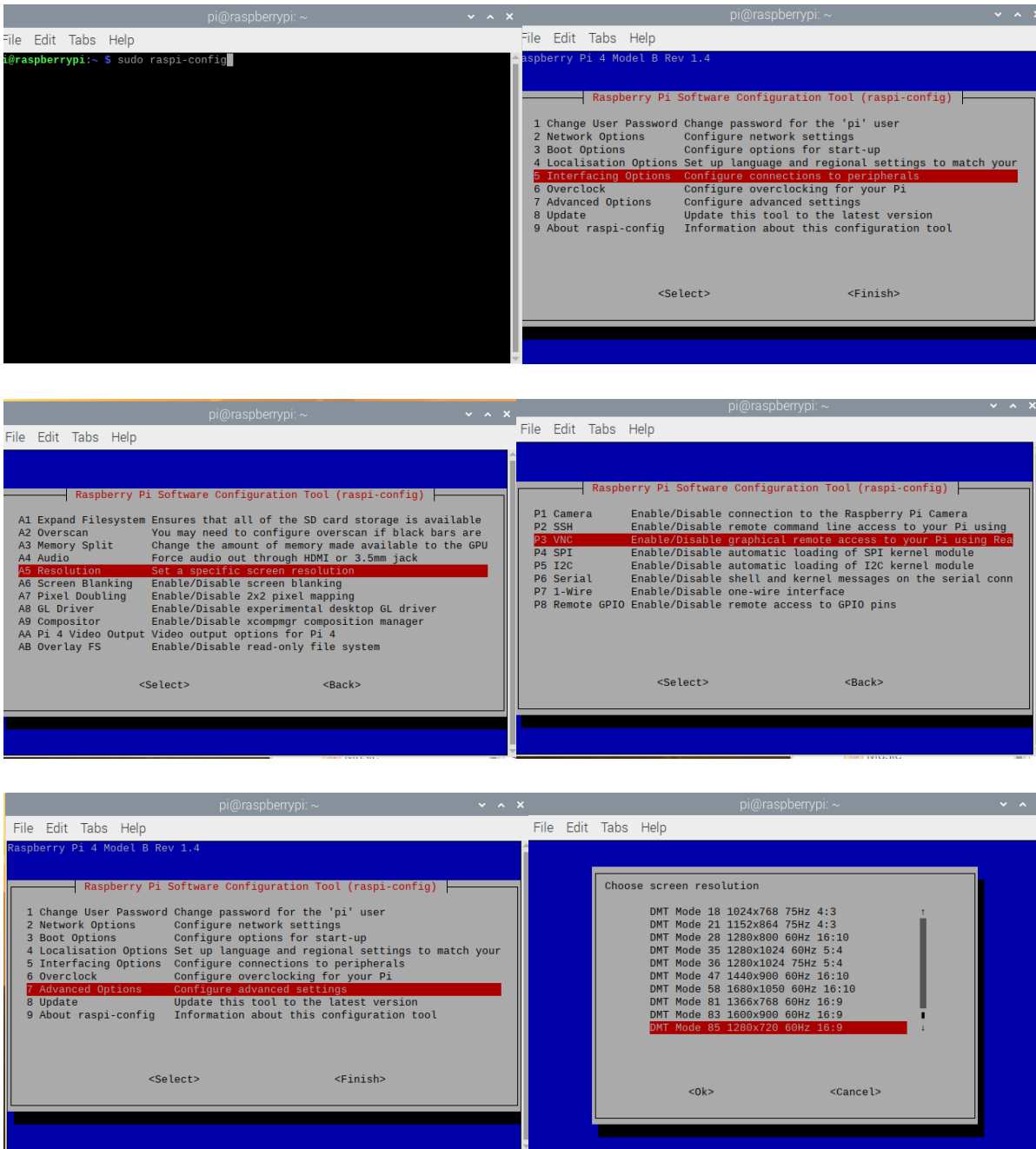


Figure 55 - Enabling the VNC viewer in Raspberry Pi and adjusting the resolution

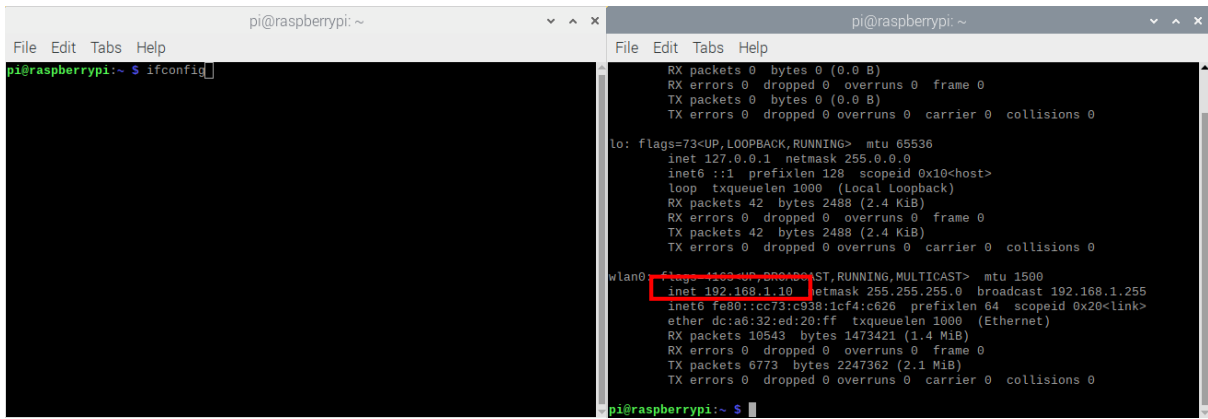


Figure 56 - Getting Raspberry Pi IP address

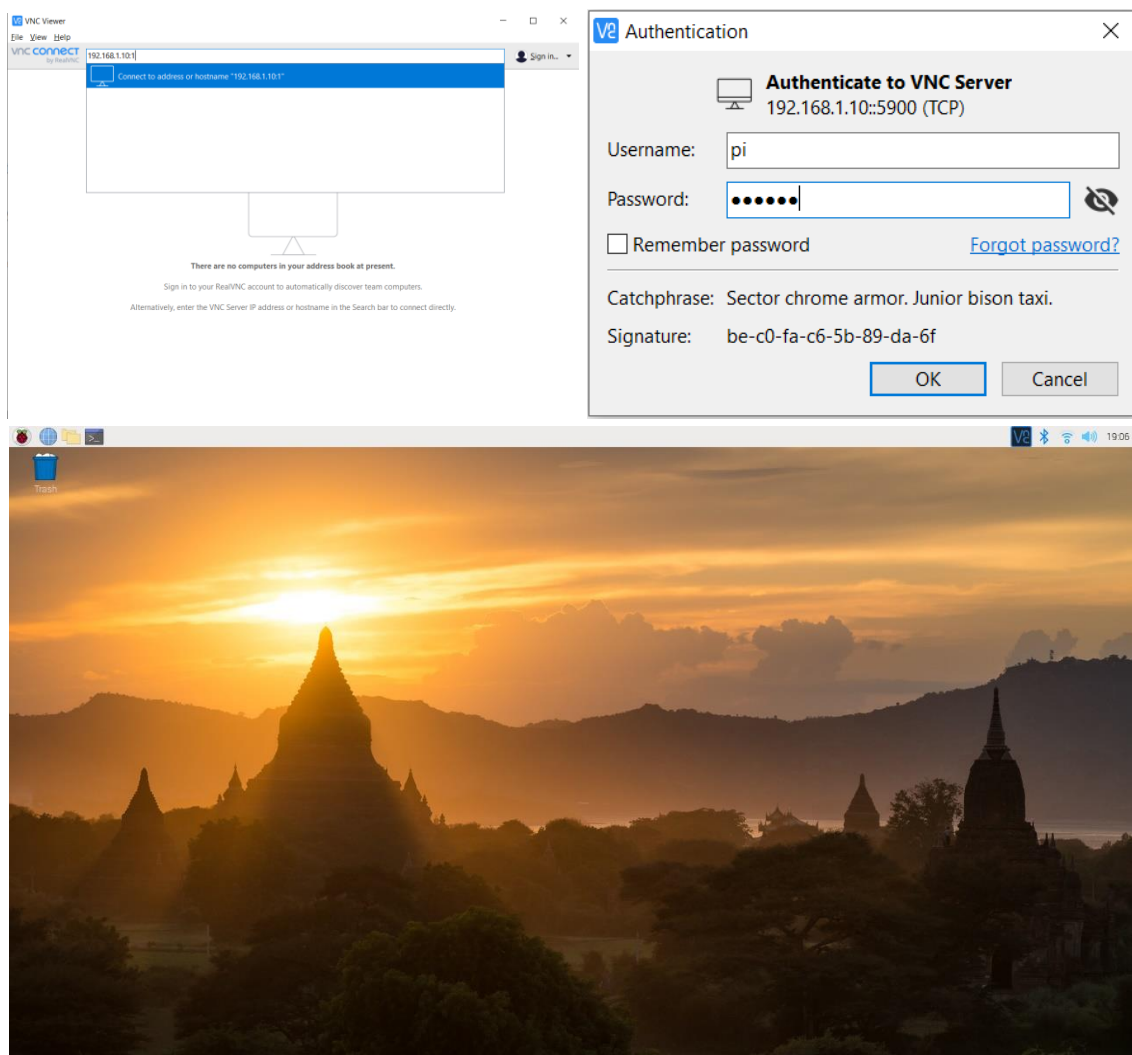


Figure 57 - Running the VNC viewer on a computer

## APPENDIX 2: SCRIPTS

```

1  #This scripts send LDR telemetries to the google sheets
2  import datetime, time
3  import gspread
4  from oauth2client.service_account import ServiceAccountCredentials
5  #Credntial
6  scope = ['https://spreadsheets.google.com/feeds','https://www.googleapis.com/auth/drive']
7  creds = ServiceAccountCredentials.from_json_keyfile_name('/home/pi/codding/mydata.json', scope)
8
9  client = gspread.authorize(creds)
10 sheet = client.open("Raspi_data").sheet1
11 #Function for reading light status from the sensor
12 def ReadLDR ():
13     import RPi.GPIO as GPIO
14     GPIO.setmode(GPIO.BCM)
15     GPIO.setwarnings(False)
16     DIGITAL_PIN = 18
17     GPIO.setup(DIGITAL_PIN, GPIO.IN)
18     try: # Main program loop
19         if GPIO.input(DIGITAL_PIN)==0:
20             print('Light detected!')
21             Status='Enough'
22         else:
23             print('No light!')
24             Status='Not Enough'
25     except KeyboardInterrupt:
26         print('\nScript end!')
27     finally:
28         GPIO.cleanup()
29     return Status
30 data = 1
31 #Function for writing the information to google sheet
32 def append_sheet(data):
33     time = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
34     print("****80)
35     values =[data,time,ReadLDR()]
36     print(values)
37     sheet.append_row(values)
38     print("Google Sheet Updated")
39     print("****80)
40 #Script body
41 while True:
42     client.login()
43     append_sheet(data)
44     data += 1
45     print("Waiting")
46     for i in range (10):
47         print(i)
48         time.sleep(1)

```

Figure 58 - Script for reading the LDR data and writing in Google Sheets

```
1 from googleapiclient.discovery import build
2 from google.oauth2 import service_account
3 import time
4 import csv
5
6 SERVICE_ACCOUNT_FILE = 'mydata.json'
7 SCOPES = ['https://www.googleapis.com/auth/spreadsheets']
8
9 creds= None
10 creds = service_account.Credentials.from_service_account_file(
11     SERVICE_ACCOUNT_FILE, scopes=SCOPES)
12
13 # The ID of a spreadsheet.
14 SAMPLE_SPREADSHEET_ID = '1jzDEOyt8BFMzoZZJModg-1qb8_P1Y2YJ7Lb8w4yPulM'
15
16 service = build('sheets', 'v4', credentials=creds)
17
18 # Call the Sheets API
19 sheet = service.spreadsheets()
20 #Script body
21 while True:
22     #reading the rows untill the end of entered data
23     result = sheet.values().get(spreadsheetId=SAMPLE_SPREADSHEET_ID,range="Light!A1:C").execute()
24     values = result.get('values', [])
25     header = values[0]
26     values.pop(0)
27     data = values
28     #Writing the data in CSV file
29     with open('Light.csv', 'w', encoding='UTF8', newline='') as f:
30         writer = csv.writer(f)
31         # write the header
32         writer.writerow(header)
33         # write multiple rows
34         writer.writerows(data)
35         f.close()
36     time.sleep(10)
```

Figure 59 - Script for reading the data from Google Sheets and store in local CSV file

```

1
2 """First you have to select the room that the sensor is installed. "last Data" mode, the last updated data will be pres
3 #this script also capable of importing temperature status and draing a chart in addition to light status
4 __title__='Lighting\n Status'
5 __author__='Meshkini'
6 import clr
7 clr.AddReferenceByPartialName('PresentationCore')
8 clr.AddReferenceByPartialName('PresentationFramework')
9 clr.AddReferenceByPartialName('System')
10 clr.AddReferenceByPartialName('System.Windows.Forms')
11 uiapp = __revit__
12 uidoc = uiapp.ActiveUIDocument
13 from pyrevit import forms
14 from Autodesk.Revit.UI import *
15 doc = uiapp.ActiveUIDocument.Document
16 from Autodesk.Revit.DB import *
17 import csv
18 import codecs
19 import os.path as op
20 from collections import defaultdict
21 from pyrevit import PyRevitException
22 from pyrevit import coreutils
23 from pyrevit import revit, DB, script
24 import sys
25 import time
26 #Reading data from CSV file
27 logger = script.get_logger()
28 def read_csv_typed_data(csv_file):
29     """Read Revit property data from the given CSV file."""
30     # open file
31     with codecs.open(csv_file, 'rb', encoding='utf-8') as csvfile:
32         # read lines
33         csv_lines = list(csv.reader(csvfile, delimiter=',', quotechar='"'))
34         field_defs = []
35         for field_def in csv_lines[0]:
36             parts = field_def.split('|')
37             parts_count = len(parts)
38             if parts_count == 1:
39                 if parts[0]:
40                     field_defs.append((parts[0], DB.ParameterType.Text))
41             elif parts_count == 2:
42                 field_defs.append((parts[0],
43                                     coreutils.get_enum_value(DB.ParameterType,
44                                                                 parts[1])))
45         # return field definitions, and data
46         return (field_defs, csv_lines[1:])
47 source_file = forms.pick_file(file_ext='csv')
48 param_defs, param_data = read_csv_typed_data(source_file)
49
50 #selecting the room(mass)
51
52 if uidoc:
53     doc = uiapp.ActiveUIDocument.Document
54     x = [doc.GetElement(x) for x in uidoc.Selection.GetElementIds()]
55     #Finding the mass parameter
56     temperature=x[0].LookupParameter("Temperature")
57     light=x[0].LookupParameter("Light")
58     lighttime= x[0].LookupParameter("Light Updated Time")
59     #creating the form
60     selected_option = forms.CommandSwitchWindow.show(
61         ['Last Data', 'Live Data',],
62         message='Select Option:',
63     )
64
65 #this function assigned a last temperature value and present the chart
66 def TempAssigneChart():
67     t=Transaction(doc, "Assigning the Temperature Value and making a chart")
68     t.Start()
69     if temperature:
70         temperature.Set(int(param_data[-1][2]))
71     t.Commit()
72     #Presenting the data on the line chart
73     output = script.get_output()
74     # get line chart object
75     chart = output.make_line_chart()
76     # this is a list of labels for the X axis of the line graph
77     chart.data.labels = []
78     for u in range(len(param_data)):
79         chart.data.labels.append(param_data[u][1])
80     # Let's add the first dataset to the chart object
81     Temperature = chart.data.new_dataset('Temperature')
82
83     # These are the data for the Y axis of the graph
84     Temperature.data = []
85     for u in range(len(param_data)):
86         Temperature.data.append(param_data[u][2])
87     # Set the color for this graph
88     Temperature.set_color(0xFF, 0x8C, 0x8D, 0.8)
89     chart.draw()
90 #this function assigned a last temperature value to the mass and print the last value
91 def TempAssigne():
92     t=Transaction(doc, "Assigning the values")
93     t.Start()
94     if temperature:

```

```

95 |         temperature.Set(int(param_data[-1][2]))
96 |         t.Commit()
97 |         print(param_data[-1][2])
98 | #this function assigned a last light status value to the mass and print the last value
99 | def LightAssigne():
100 |     t=Transaction(doc, "Assigning the values")
101 |     t.Start()
102 |     if light:
103 |         light.Set(param_data[-1][2])
104 |         t.Commit()
105 |         print(param_data[-1][2])
106 | #this function assigned a last updated time of the light status value to the mass and print the last value
107 | def LightDateAssigne():
108 |     t=Transaction(doc, "Assigning the values")
109 |     t.Start()
110 |     if lighttime:
111 |         lighttime.Set(param_data[-1][1])
112 |         t.Commit()
113 |         print(param_data[-1][1])
114 | #changing the transparency of the room(mass) >> 100: Transparent, 0:not transparent
115 | def RoomTransparency(El, Tr):
116 |     t=Transaction(doc, "ChangeTransparency")
117 |     t.Start()
118 |     NewGraphic= OverrideGraphicSettings()
119 |     NewGraphic.SetSurfaceTransparency(Tr)
120 |     doc.ActiveView.SetElementOverrides(El.Id, NewGraphic)
121 |     t.Commit()
122 |
123 |
124 | #presenting info in the "Last Data" mode
125 | if selected_option == 'Last Data':
126 |     #TempAssigneChart()
127 |     LightAssigne()
128 |     LightDateAssigne()
129 |     if param_data[-1][2]=='Not Enough':
130 |         RoomTransparency(x[0],5)
131 |     else:
132 |         RoomTransparency(x[0],95)
133 |
134 |
135 | #presenting info in the "Live Data" mode
136 | if selected_option == 'Live Data':
137 |     for r in range(3):
138 |         #TempAssigne()
139 |         LightAssigne()
140 |         if param_data[-1][2]=='Not Enough':
141 |             RoomTransparency(x[0],5)
142 |         else:
143 |             RoomTransparency(x[0],95)
144 |         time.sleep(10)
145 |
146 |

```

Figure 60 - The script of "Receiving" panel written in pyRevit

```

1  |
2  | L*****First you have to select the window that the motor is installed. "Open" mode, will open the Window completely. "Close" mode will c]
3  |
4  |     _title_ ='Windows\n Automation'
5  |     _author_ ='Meshkini'
6  |
7  | import clr
8  | clr.AddReferenceByPartialName('PresentationCore')
9  | clr.AddReferenceByPartialName('PresentationFramework')
10 | clr.AddReferenceByPartialName('System')
11 | clr.AddReferenceByPartialName('System.Windows.Forms')
12 | uiapp = __revit__
13 | uidoc = uiapp.ActiveUIDocument
14 | from pyrevit import forms
15 | from Autodesk.Revit.UI import *
16 | doc = uiapp.ActiveUIDocument.Document
17 | from Autodesk.Revit.DB import *
18 | import csv
19 | import codecs
20 | import os.path as op
21 | from collections import defaultdict
22 | from pyrevit import PyRevitException
23 | from pyrevit import coreutils
24 | from pyrevit import revit, DB, HOST_APP
25 | from pyrevit import script
26 | import sys
27 | import time, math, datetime
28 | import Autodesk
29 | import System
30 | import math
31 | view=doc.ActiveView
32 |
33 | logger = script.get_logger()
34 |
35 | #this function sets the sun position in a real position based on the time of request
36 | def SunPosUpdate():
37 |     output = script.get_output()
38 |     # Close other output when launched several times
39 |     close_other_output = output.close_others(all_open_outputs=True)
40 |     doc = revit.doc
41 |     view = doc.ActiveView
42 |     uidoc = HOST_APP.uidoc
43 |     DT = System.DateTime.Now
44 |
45 |     with revit.Transaction('Set sun Date and Time to Now'):
46 |         settings_sun = view.SunAndShadowSettings
47 |         settings_sun.StartDateAndTime = DT
48 |         uidoc.RequestViewChange(view)
49 |         uidoc.RefreshActiveView()
50 |         doc.Regenerate()
51 |
52 |     with revit.Transaction('Touch and Go'):
53 |         sun_path_size = settings_sun.get_Parameter(DB.BuiltInParameter.VIEW_GRAPH_SUN_PATH_SIZE).AsInteger()
54 |         temp_sun_path_size = sun_path_size
55 |         param = settings_sun.Parameter[DB.BuiltInParameter.VIEW_GRAPH_SUN_PATH_SIZE]
56 |         param.Set(10)
57 |         param.Set(temp_sun_path_size)
58 |
59 | #selecting the windows
60 |
61 | if uidoc:
62 |     doc = uiapp.ActiveUIDocument.Document
63 |     x = [doc.GetElement(x) for x in uidoc.Selection.GetElementIds()]
64 |
65 |     Angle=x[0].LookupParameter("SwingAngle")
66 |
67 | #Assign value to the window rotation
68 | def Assigne(a):
69 |     t=Transaction(doc, "Assigning the values")
70 |     t.Start()
71 |     if Angle:
72 |         Angle.Set(math.radians(a))
73 |     t.Commit()
74 |
75 |
76 | #Export rotation angle to the CSV
77 | def csvExpo(b):
78 |     a=[datetime.datetime.now()]
79 |     a.append(b)
80 |     csv_rows = [a]
81 |     csv_file_path = "D:\peoplel.csv"
82 |     file = script.dump_csv(csv_rows, csv_file_path)
83 | #calculate the direction of sun with the north axis
84 | def SunDirection():
85 |     t=Transaction(doc, "Assigning the values")
86 |     t.Start()
87 |     j=datetime.datetime.now()
88 |     Dt_kind=System.DateTimeKind.Local
89 |     specTime=System.DateTime.SpecifyKind(j,Dt_kind)
90 |     settings=view.SunAndShadowSettings
91 |     settings.SunAndShadowType = SunAndShadowType.StillImage
92 |     uidoc.RefreshActiveView()
93 |     doc.Regenerate()
94 |     e=(settings.GetFrameAzimuth(settings.ActiveFrame))*180/math.pi

```

```

95     if e<0:
96         e=360+e
97     t.Commit()
98     return round(e)
99
100 #calculate the direction of windows with the north axis- angle between normal vector of the window and north(clockwise is positive
101 def WinDirection():
102     if uidoc:
103         doc = uiapp.ActiveUIDocument.Document
104         x = [doc.GetElement(x) for x in uidoc.Selection.GetElementIds()]
105         s=x[0].Host
106         WinOrX=s.Orientation.X
107         WinOrY=s.Orientation.Y
108         WinInAngle=math.asin(abs(s.Orientation.X))*180/math.pi
109         if WinOrX<0 and WinOrY<0:
110             WinFinalAngle = WinInAngle
111         if WinOrX<0 and WinOrY>0:
112             WinFinalAngle = 180-WinInAngle
113         if WinOrX>0 and WinOrY>0:
114             WinFinalAngle = WinInAngle+180
115         if WinOrX>0 and WinOrY<0:
116             WinFinalAngle = 360-WinInAngle
117         return round(WinFinalAngle)
118 #this function calculates the angle between the window louver and north axis
119 def LouverAngle():
120     l=-1*(Angle.AsDouble()*180/math.pi)+WinDirection()
121     return l
122 # create the form
123 selected_option = forms.CommandSwitchWindow.show(
124     ['Open', 'Close', 'Summer Mode (Current time)', 'Winter Mode (Current time)', 'Summer Mode (Revit time)', 'Winter Mode (Revit time)'],
125     message='Select Option:',
126 )
127
128 if selected_option == 'Open':
129     ClockRotation=round(90+(Angle.AsDouble()/((math.pi)/180)))
130     Assigne(-90)
131     csvExpo(ClockRotation)
132
133
134 if selected_option == 'Close':
135     ClockRotation=round((Angle.AsDouble()/((math.pi)/180)))
136     Assigne(0)
137     csvExpo(ClockRotation)
138
139 if selected_option == 'Summer Mode (Current time)':
140     SunPosUpdate()
141     ExpectedClockRotation=SunDirection()-WinDirection()
142     ExpectedClockRotation=SunDirection()-WinDirection()
143     ClockRotation=ExpectedClockRotation+(Angle.AsDouble()*180/math.pi)
144     if ClockRotation+LouverAngle()>=90+WinDirection():
145         ClockRotation=round(90+(Angle.AsDouble()*180/math.pi))
146         Assigne(-90)
147     elif ClockRotation+LouverAngle()<=-90+WinDirection():
148         ClockRotation=round(-90+(Angle.AsDouble()*180/math.pi))
149         Assigne(90)
150     else:
151         s=ExpectedClockRotation
152         ClockRotation=ExpectedClockRotation+round(Angle.AsDouble()/((math.pi)/180))
153         Assigne(-s)
154     csvExpo(ClockRotation)
155
156 if selected_option == 'Winter Mode (Current time)':
157     SunPosUpdate()
158     ExpectedClockRotation=SunDirection()-WinDirection()+90
159     ClockRotation=ExpectedClockRotation+(Angle.AsDouble()*180/math.pi)
160     if ClockRotation+LouverAngle()>=90+WinDirection():
161         ClockRotation=round(90+(Angle.AsDouble()*180/math.pi))
162         Assigne(-90)
163     elif ClockRotation+LouverAngle()<=-90+WinDirection():
164         ClockRotation=round(-90+(Angle.AsDouble()*180/math.pi))
165         Assigne(90)
166     else:
167         s=ExpectedClockRotation
168         ClockRotation=ExpectedClockRotation+round(Angle.AsDouble()/((math.pi)/180))
169         Assigne(-s)
170     csvExpo(ClockRotation)
171
172
173
174
175 if selected_option == 'Summer Mode (Revit time)':
176     ExpectedClockRotation=SunDirection()-WinDirection()
177     ClockRotation=ExpectedClockRotation+(Angle.AsDouble()*180/math.pi)
178     if ClockRotation+LouverAngle()>=90+WinDirection():
179         ClockRotation=round(90+(Angle.AsDouble()*180/math.pi))
180         Assigne(-90)
181     elif ClockRotation+LouverAngle()<=-90+WinDirection():
182         ClockRotation=round(-90+(Angle.AsDouble()*180/math.pi))
183         Assigne(90)
184     else:
185         s=ExpectedClockRotation
186         ClockRotation=ExpectedClockRotation+round(Angle.AsDouble()/((math.pi)/180))
187         Assigne(-s)
188     csvExpo(ClockRotation)
189
190
191 if selected_option == 'Winter Mode (Revit time)':
192     ExpectedClockRotation=SunDirection()-WinDirection()+90
193     ClockRotation=ExpectedClockRotation+(Angle.AsDouble()*180/math.pi)
194     if ClockRotation+LouverAngle()>=90+WinDirection():
195         ClockRotation=round(90+(Angle.AsDouble()*180/math.pi))
196         Assigne(-90)
197     elif ClockRotation+LouverAngle()<=-90+WinDirection():
198         ClockRotation=round(-90+(Angle.AsDouble()*180/math.pi))
199         Assigne(90)
200     else:
201         s=ExpectedClockRotation
202         ClockRotation=ExpectedClockRotation+round(Angle.AsDouble()/((math.pi)/180))
203         Assigne(-s)
204     csvExpo(ClockRotation)
205

```

**Figure 61 - Script for modifying the model based on the selected scenario, getting rotation data from Revit, and writing in the local CSV file**

```
1 import csv
2 import time
3 import gspread
4 from oauth2client.service_account import ServiceAccountCredentials
5
6 #reading the CSV file (Rotation Angle)
7 def ReadCSV ():
8     f=open('D:\Rotation.csv')
9     csv_f=csv.reader(f)
10    for row in csv_f:
11        print (row)
12    f.close()
13    return (row)
14
15 #Writing the data in Google sheet
16 def GoogleSheetWrit (c):
17     scope = ['https://spreadsheets.google.com/feeds','https://www.googleapis.com/auth/drive']
18     creds = ServiceAccountCredentials.from_json_keyfile_name('mydata.json', scope)
19     client = gspread.authorize(creds)
20     sheet = client.open("Raspi_data").worksheet('RotationAngle')
21     sheet.append_row(c)
22 #read the last data from googlesheet
23 def ReadLast ():
24     scope = ['https://spreadsheets.google.com/feeds','https://www.googleapis.com/auth/drive']
25     creds = ServiceAccountCredentials.from_json_keyfile_name('mydata.json', scope)
26     client = gspread.authorize(creds)
27     sheet = client.open("Raspi_data").worksheet('RotationAngle')
28     return (sheet.get_all_values() [-1])
29 #Main body, Reading data from local csv file, and writing in google sheet and also control not to write repetitive data
30 while True:
31     if ReadLast() == ReadCSV ():
32         print ('No new data')
33     else:
34         GoogleSheetWrit (ReadCSV ())
35     time.sleep(10)
```

Figure 62 - Python script for transferring data from a local CSV file to the Google Sheets

```

1 import datetime, time
2 import gspread
3 from oauth2client.service_account import ServiceAccountCredentials
4 import RPi.GPIO as GPIO
5 import math
6 scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']
7 creds = ServiceAccountCredentials.from_json_keyfile_name('/home/pi/codding/mydata.json', scope)
8 client = gspread.authorize(creds)
9 #Assigning the pin numbers
10 in1 = 17
11 in2 = 18
12 in3 = 27
13 in4 = 22
14
15 #Reading the data from googlesheet
16 def ReadGoogleSheet():
17     WorkSheet = client.open("Raspi_data").worksheet("RotationAngle")
18     Values=WorkSheet.get_all_values()
19     print (Values)
20     return (Values)
21
22
23 def cleanup():
24
25     GPIO.output( in1, GPIO.LOW )
26     GPIO.output( in2, GPIO.LOW )
27     GPIO.output( in3, GPIO.LOW )
28     GPIO.output( in4, GPIO.LOW )
29     GPIO.cleanup()
30 #this function send a command to the motor for rotating
31 def MotorMove(A):
32     step_sleep = 0.002
33
34     step_count = round(abs(A)/360*4096) # 5.625*(1/64) per step, 4096 steps is 360°
35
36     if A>0:
37         direction = True # True for clockwise, False for counter-clockwise
38     else:
39         direction = False
40
41     # defining stepper motor sequence
42     step_sequence = [[1,0,0,1],
43                     [1,0,0,0],
44                     [1,1,0,0],
45                     [0,1,0,0],
46                     [0,1,1,0],
47                     [0,0,1,0],
48                     [0,0,1,1],
49                     [0,0,0,1]]
50
51     # setting up
52     GPIO.setmode( GPIO.BCM )
53     GPIO.setup( in1, GPIO.OUT )
54     GPIO.setup( in2, GPIO.OUT )
55     GPIO.setup( in3, GPIO.OUT )
56     GPIO.setup( in4, GPIO.OUT )
57
58     # initializing
59     GPIO.output( in1, GPIO.LOW )
60     GPIO.output( in2, GPIO.LOW )
61     GPIO.output( in3, GPIO.LOW )
62     GPIO.output( in4, GPIO.LOW )
63
64
65     motor_pins = [in1,in2,in3,in4]
66     motor_step_counter = 0 ;
67
68
69     try:
70         i = 0
71         for i in range(step_count):
72             for pin in range(0, len(motor_pins)):
73                 GPIO.output( motor_pins[pin], step_sequence[motor_step_counter][pin] )
74             if direction==True:
75                 motor_step_counter = (motor_step_counter - 1) % 8
76             elif direction==False:
77                 motor_step_counter = (motor_step_counter + 1) % 8
78             else: # defensive programming
79                 print( "uh oh... direction should *always* be either True or False" )
80                 cleanup()
81                 exit( 1 )
82             time.sleep( step_sleep )
83
84     except KeyboardInterrupt:
85         cleanup()
86         exit( 1 )
87
88     cleanup()
89     print(round(abs(A)/360*4096))
90
91 PreCommands=ReadGoogleSheet()[0]
92 #main body of script
93 while True:
94     RGSH=ReadGoogleSheet()
95     if len(ReadGoogleSheet())>1:
96         if ReadGoogleSheet()[0]==PreCommands:
97             print ("No new command")
98         else:
99             print("new command")
100             MotorMove(int(RGSH[-1][1]))
101             PreCommands=ReadGoogleSheet()[0]
102     else:
103         print ("No new command")
104
105     time.sleep(10)

```

**Figure 63 - Reading the commands from Google Sheets and sending to the motor via Python script in Raspberry Pi**